DeviceAnywhere™
by Mobile Complete

DeviceAnywhere™ User Guide

©2008 Mobile Complete, Inc. All rights reserved.

THE INFORMATION CONTAINED IN THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT ANY EXPRESS REPRESENTATIONS OF WARRANTIES. IN ADDITION, MOBILE COMPLETE, INC., DISCLAIMS ALL IMPLIED REPRESENTATIONS AND WARRANTIES, INCLUDING ANY WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTURAL PROPERTY RIGHTS.

DeviceAnywhere Studio Version 4.2

All text and figures included in this publication are the exclusive property of Mobile Complete Inc., and may not be copied, reproduced, or used in any way without the express permission in writing of Mobile Complete Inc. Information in this document is subject to change without notice and does not represent a commitment on the part of Mobile Complete, Inc. Mobile Complete, Inc may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents except as expressly provided in any written license agreement from Mobile Complete, Inc.

Mobile Complete, the Mobile Complete logo, DeviceAnywhere, DeviceAnywhere Studio, DeviceAnywhere Pro, DeviceConductor and MyDeviceAnywhere are trademarks of Mobile Complete, Inc. in the United States and/or other countries.

This document also contains registered trademarks, trademarks and service marks that are owned by their respective owners. Mobile Complete, Inc. disclaims any responsibility for specifying marks that are owned by their respective companies or organizations. If you have any comments or suggestions regarding this document, please send them by e-mail to DAsupport@deviceanywhere.com.

Mobile Complete, Inc.
1730 South Amphlett Blvd., STE. 300
San Mateo, CA 94402 USA

Table of Contents

List of Figures 10
List of Tables 15
Introduction 16
1.1 *DeviceAnywhere* System Requirements 17
1.2 About This User Guide 17
1 *DeviceAnywhere Studio* Interface 20
1.1 Command Entry 21
1.2 *DeviceAnywhere* Help 22
1.3 *DeviceAnywhere* Unexpected Termination 22
2 Getting Started 24
2.1 Log in to *DeviceAnywhere* *24*
2.2 *MyDeviceAnywhere* Website 25
2.3 Launch *DeviceAnywhere Studio* *25*
2.4 Log in to *DeviceAnywhere Studio* 26
3 Device Access 27
3.1 *DeviceAnywhere* System Setup 27
3.1.1 Change the Resolution of your Monitor 27
3.1.2 Set the Color Depth of the Monitor 28
3.2 Device Access Overview 28
3.2.1 Device Menu 29
3.2.2 On-Screen Icons 31
3.3 Device Session 32
3.3.1 Access a Device 32
3.3.2 Device Window 32
3.3.3 Device Availability and Status 32
3.3.4 Wait for a Device 33
3.3.5 Release a Device 33
3.3.6 Reset Device 33
3.3.7 Wake Up a Device 33
3.3.8 Run Cleanup Scripts 34
3.4 Device Properties 35
3.5 Device Video Controls 36
3.6 Interactive Session 37
3.6.1 Start an Interactive Session 37
3.7 Key Press Functions 38
3.7.1 Key Modes 38
3.7.2 Copy and Paste Commands 38
3.7.3 Holding Down a Key on a Device 39
3.7.4 Multiple Key Mapping 39
3.8 Device Audio Controls 40
3.9 Hardware Control 41
3.10 Device Collaboration 41
3.10.1 Sharing a Device 41
3.10.2 Inviting Users 42
3.10.3 Accepting an Invitation 43

3.10.4 Stop Observing a Device 44
3.10.5 Sending Messages 44
3.11 Exporting Frames and Video Media 46
3.11.1 Video History 47
3.11.2 Export Commands 47
3.11.3 Export the Current Frame 48
3.11.4 Upload Frames 48
3.12 Viewing Uploaded Frames in *MyDeviceAnywhere* *49*
3.13 Managing Wireless Applications 51
3.13.1 BREW Devices 52
4 Record & Playback 54
4.1 Record & Playback Overview 54
4.2 Get Ready to Record a Script 55
4.2.1 Manage Folders and Scripts 55
4.2.2 Add a Device to the Script 56
4.2.3 Access and Acquire a Device 56
4.3 Script Commands 58
4.3.1 SendKeys Command 58
4.3.2 WaitImage Command 59
4.4 Create an End-to-End Script 60
4.4.1 Record and Playback Controls 61
4.4.2 Sample End-to-End Script—Set Alarm 61
4.4.3 What Your Automated Script Looks Like 64
4.4.4 Working with Scripts 65
4.4.5 Script Status 65
4.4.6 Open the Script Menu 65
4.4.7 Perform Simple Tasks with the Script Menu 65
4.4.8 Script Properties 66
4.4.9 Properties Screen Toolbar 67
4.4.10 Default Test Environment 67
4.4.11 Script History 67
4.4.12 Script File Properties 68
4.4.13 Dataset 69
4.5 SendKeys Properties 70
4.5.1 SendKeys Properties 71
4.5.2 WaitImage Tab in SendKeys Properties 71
4.5.3 Timeout Tab in SendKeys Properties 72
4.5.4 WaitImage Properties 74
4.5.5 Set the Reference Image 75
4.6 Play Script 75
4.7 Upload Test Results 75
5 Automation 77
5.1 Automation Overview 77
5.1.1 Automation View 78
5.2 Get Ready to Build a Test Script 79
5.2.1 Use a Project 79
5.2.2 Create a Folder within a Project 80
5.2.3 Create a Script 80
5.2.4 File Naming Conventions 81
5.2.5 Add a Device to the Script 82
5.2.6 Access and Acquire a Device 82
5.3 Multi-Device Scripts 83

5.4 Object-Oriented Scripting and Record and Playback 84
5.4.1 Adding Script Commands to the Script 84
5.4.2 SendKeys Command Example 84
5.4.3 SendKeys Properties 85
5.4.4 WaitImage Tab in SendKeys Properties 86
5.4.5 Timeout Tab in SendKeys Properties 87
5.4.6 Enter Text in the Text to Send Field 88
5.4.7 Labeling Test Steps 89
5.4.8 Running a Script with Run from Here 90
5.5 Script Editor 90
5.5.1 Record and Playback Controls 90
5.6 Working with Scripts 91
5.6.1 Script Status 91
5.6.2 Open the Script Menu 91
5.6.3 Perform Simple Tasks with the Script Menu 92
5.6.4 Script Properties Screen 93
5.6.5 Properties Screen Toolbar 93
5.6.6 Default Test Environment 94
5.6.7 Script History 94
5.6.8 Script File Properties 94
5.6.9 Dataset 95
5.7 Recording a Test Script 96
5.7.1 Prepare for Recording 96
5.7.2 Recording the Test Script 97
5.7.3 Saving Changes 99
5.7.4 Playing the Test Script 100
5.7.5 Uploading Test Results 100
5.8 Building a Send SMS Script 102
5.8.1 Create the Send SMS Test Procedure 103
5.8.2 Create the Test Procedure 104
5.8.3 Create the Macro Resources 104
5.8.4 Create Macro Instances—Reset to Idle 106
5.8.5 Reset to Idle—Macro Instance Summary 111
5.9 Create Macro Instances—Send SMS 114
5.9.1 Return to Idle 116
5.9.2 Verify Idle 117
5.9.3 Nav to Messaging 118
5.9.4 Verify Messaging 118
5.9.5 Select New Message 119
5.9.6 Insert Phone Number 119
5.9.7 Insert TXT Message 119
5.9.8 Send TXT Message 119
5.9.9 Verify Message Sent 120
5.9.10 Return to Idle 121
5.9.11 Success 121
5.9.12 Set Up Macro Parameters in Send SMS 121
5.9.13 Insert Local Variables in the SendKeys 123
5.9.14 Associate Macro Resources in the Test Procedure 123
5.9.15 Save Changes to Send Message Script 125
5.9.16 Play Your New Send Message Script 125
5.10 Work with Variables in a Script 126
5.10.1 Variable Features 126
5.10.2 DataSet Tab in Script Properties 127

5.10.3 Create Variables for a Dataset 128
5.10.4 Implementing Variables in a Script 128
5.10.5 Check How to Use Records 131
5.10.6 Create Macro Parameters 131
5.10.7 Create Variables in a Macro Resource 132
5.10.8 Create Global Variables 134
5.10.9 Set Up a Global Variable 134
5.10.10 Implement Global Variables in a Test Script 134
5.10.11 Associate a Global Variable 135
5.11 Test Suites 136
5.11.1 Test Suite View 136
5.11.2 Creating a Test Suite 137
5.11.3 Adding Scripts to a Test Suite 138
5.11.4 Configuring a Script or Macro 138
5.12 Playing a Test Suite 140
5.12.1 Viewing Test Suite Results 141
5.12.2 Sharing Results 142
5.13 Scheduler 143
5.13.1 Schedule Editor Toolbar 144
5.13.2 Scheduling Test Suites 144
5.13.3 Create a Device Bundle 145
5.13.4 Viewing the Schedule Calendar 146
5.14 Viewing Scheduled Test Suite Results 147
5.15 Dashboard 148
5.15.1 *DeviceAnywhere Pro* Servers Menu 148
5.15.2 Dashboard Overview 149
5.15.3 Preview Runs 149
5.15.4 Stop Preview Run 149
5.15.5 Delete Preview Run 149
6 Monitoring 150
6.1 Monitoring Overview 150
6.1.1 Monitoring Workflow 151
6.2 Log in to *DeviceAnywhere Monitoring* *151*
6.3 Monitoring View 153
6.4 Get Ready to Build a Monitoring Script 154
6.4.1 Use a Project 154
6.4.2 Create a Folder within a Project 155
6.4.3 Create a Monitoring Script 155
6.4.4 Add a Device to the Script 156
6.4.5 Access and Acquire a Device 157
6.5 Multi-Device Scripts 158
6.6 Scheduler 159
6.6.1 Creating a Production Monitor 160
6.6.2 Monitor Status 160
6.6.3 Open the Production Monitor Menu 160
6.6.4 Monitor Script and Device Bundle 161
6.6.5 Create a Device Bundle 162
6.6.6 Execution Frequency 163
6.6.7 Service Level Agreements 165
6.6.8 SLA Overview 165
6.6.9 Error Management 166
6.6.10 Error Categories 166
6.6.11 Error Types 167

6.6.12 Severity 169
6.6.13 Performing Simple Tasks with Error Types and Categories 169
6.6.14 SLA Types 170
6.6.15 Create an SLA 170
6.6.16 Manage SLA Policies 177
6.6.17 Action Path 177
6.6.18 Escalation 178
6.6.19 Resolution 180
6.6.20 Execute Production Monitor 181
6.7 Dashboard 182
6.7.1 Global Reconnect 184
6.7.2 Configure 184
6.7.3 Find 184
6.8 Preview Runs 184
6.9 Video Monitor 187
6.9.1 Video Monitor Overview 187
6.9.2 General KPI Group 187
6.9.3 Video Monitor Ad Hoc Testing 188
6.9.4 Add Device to KPI Group 188
6.9.5 Access and Acquire a Device 188
6.9.6 Video Monitor 189
6.9.7 Add a Device and Acquire the Device 189
6.9.8 Start a Monitor Session 191
6.9.9 Viewing an SLA Violation 191
6.9.10 Composite Monitor Views 192
6.9.11 VM Commands in Scripts 193
6.10 Object-Oriented Scripting 193
6.10.1 Adding Script Commands to the Script 193
6.10.2 SendKeys Command Example 194
6.10.3 Recording the Key Presses 195
6.10.4 Entering Text Manually in Text to Send Field 196
6.10.5 Labeling Test Steps 196
6.10.6 Running a Script with Run from Here 197
6.11 Script Editor 197
6.11.1 Record and Playback Controls 197
6.12 Working with Scripts 198
6.12.1 Script Status 198
6.12.2 Open the Script Menu 198
6.12.3 Perform Simple Tasks with the Script Menu 199
6.12.4 Monitor Script Properties Screen 199
6.12.5 Properties Screen Toolbar 200
6.12.6 Default Test Environment 200
6.12.7 Script History 201
6.12.8 Script File Properties 201
6.12.9 Dataset 202
6.13 Send Message Script with Timers 203
6.14 Building an ESPN News Monitoring Script 207
6.14.1 Create New Script and Acquire a Device 212
6.14.2 Clear Macro 213
6.14.3 Clear Cache 213
6.14.4 Menu 215
6.14.5 Cingular Video 215
6.14.6 Start Launch CV 216

6.14.7 Wait Video 217
6.14.8 Set the Reference Image 219
6.14.9 Stop Launch CV 219
6.14.10 Save Changes to the ESPN News Script 219
6.14.11 Play Your New ESPN News Script 220
6.14.12 Upload Test Results 220
6.14.13 Viewing Production Monitoring Results 221
7 *DeviceAnywhere* Script Commands 223
7.1 Script Commands Overview 223
7.2 CheckPoint 223
7.3 Script Command Descriptions 225
7.3.1 SendKeys Command 225
7.3.2 SendKeys Properties Screen 225
7.3.3 SendKeys Properties 226
7.3.4 WaitImage Tab in SendKeys Properties 227
7.3.5 Timeout Tab in SendKeys Properties 228
7.3.6 Checkpoint Tab in SendKeys Properties 230
7.3.7 Hardware Command 231
7.3.8 PlayAudio 231
7.3.9 FindAndTouch 232
7.3.10 Wait Command 233
7.3.11 WaitText 234
7.3.12 Using WaitText 234
7.3.13 WaitImage Command 235
7.3.14 WaitImage Properties 236
7.3.15 Set the Reference Image 237
7.3.16 WaitEvent Command 237
7.3.17 WaitAudio Command 238
7.3.18 NavigateTo Command 239
7.3.19 SetVariable Command 241
7.3.20 Branch Command 242
7.3.21 Loop Command 244
7.3.22 Toggle Timers 245
7.3.23 Toggle Timer Properties Screen 245
7.3.24 Creating Timers 245
7.3.25 VM Command 246
7.3.26 Macro Command 247
7.3.27 Java Command 248
7.3.28 Success Command 250
7.3.29 Fail Command 250
9 Resource Manager 252
9.1 Resource Manager Overview 252
9.2 What is a Resource? 253
9.2.1 Macro Resources 254
9.2.2 Design Efficient Resources 254
9.2.3 Create Resources in Two Ways 255
9.3 Setting up a Resource Library 255
9.3.1 Add a Resource Library 255
9.3.2 Add Category Folders 256
9.3.3 Create a Resource 256
9.3.4 Create a Resource Instance 257
9.3.5 Implementing a Resource 258
9.3.6 Edit a Resource Instance 259

9.4 Resource Description 259
9.5 Creating a Macro Library and Macros 260
9.5.1 Add a Resource Library 260
9.5.2 Add a Category Folder 261
9.5.3 Add a Macro 261
9.5.4 Add a Macro Instance 261
9.5.5 Access and Acquire the Device 262
9.5.6 Implement a Macro Instance 262
9.5.7 Manage a Macro Instance 263
9.5.8 Macro Instance Examples 263
9.6 Bring a Resource into a Script 268
9.6.1 Bring a Macro into a Script 269
9.6.2 Create a Resource On-the-Fly 270
9.6.3 Create a Macro On-the-Fly 271
9.7 Designing Scripts with Resources 272
9.7.1 Send SMS Message with Resources—An Example 272
9.7.2 Script for Send SMS Message 272
9.7.3 Key Sequence Comparison 276
9.7.4 Final Check for Send an SMS Message 278
10 *MyDeviceAnywhere* *279*
10.1 Log in to Your *MyDeviceAnywhere* Account 279
10.2 *MyDeviceAnywhere* Interface 279
10.3 Access Tabs for Users 281
10.3.1 My Account 281
10.3.2 Subscriptions 281
10.3.3 Editing Your Subscriptions 282
10.3.4 Account Info 284
10.3.5 Usage History 284
10.3.6 Invoices 285
10.3.7 Project list 285
10.3.8 Uploaded Frames 286
10.3.9 My Applications 288
10.3.10 Test Results 289
10.3.11 Reservations 290
10.3.12 Device Search 292
10.3.13 Production Monitoring 293
10.4 Access Tabs for System Admin 295
10.4.1 Add a Package 296
10.4.2 Add Devices to a Package 297
10.4.3 Create a User Group 298
10.4.4 Add a User 299
10.4.5 Devices 300
10.4.6 Add a Project 300
10.4.7 DA Pro 301
10.5 Assign *DeviceAnywhere Pro* Licenses 301
10.6 Access Tabs for Account Admin 302
10.6.1 Account Profile 302
10.6.2 Usage History 302

List of Figures

Figure 1-1 Device Access View 20
Figure 1-2 Recovered Devices 23
Figure 2-1 *DeviceAnywhere* Login 24
Figure 2-2 *MyDeviceAnywhere* Home Page 25
Figure 2-3 Device Access View 26
Figure 3-1 Device Access View 29
Figure 3-2 Device Menu 30
Figure 3-3 On-Screen Icons 31
Figure 3-4 Wake Up Device Prompt Screen 34
Figure 3-5 Wake Up Device 34
Figure 3-6 Device Properties 35
Figure 3-7 Acquired Device with Video Control Bar 36
Figure 3-8 Multiple Key Mapping Screen 39
Figure 3-9 Buffer Properties Screen 40
Figure 3-10 On-Screen Icon for Share Device 41
Figure 3-11 Share Device Dialog Box 42
Figure 3-12 Stop Monitoring (Top right) 44
Figure 3-13 Message Window 45
Figure 3-14 Frame Viewer 46
Figure 3-15 Export Command 47
Figure 3-16 Video from History Dialog 48
Figure 3-17 Upload Frame Details 50
Figure 3-18 Email Form Letter 51
Figure 3-19 BREW Menu 52
Figure 4-1 Record and Playback View 54
Figure 4-2 Empty Script 55
Figure 4-3 Add to Script 56
Figure 4-4 Acquire Device Menu 57
Figure 4-5 Record & Playback Script—Ready to Record 58
Figure 4-6 Empty Script 60
Figure 4-7 Set Alarm Script 63
Figure 4-8 Script Properties Screen 66
Figure 4-9 History Tab in Script Properties 68
Figure 4-10 General Tab in Script Properties 69
Figure 4-11 Dataset Tab in Script Properties 69
Figure 4-12 SendKeys Properties Screen 70
Figure 4-13 WaitImage Tab in SendKeys Properties 71
Figure 4-14 Timeout Tab 72
Figure 4-15 WaitImage Properties Screen 74
Figure 4-16 Script Status Screen 75
Figure 5-1 Automation View 78
Figure 5-2 DA-Project 79
Figure 5-3 New Folder Option 80
Figure 5-4 Send SMS Script 80
Figure 5-5 Empty Script in the Script Editor 81

Figure 5-6 Add to Script 82
Figure 5-7 Access and Acquire a Device 82
Figure 5-8 SendKeys Command Dragged to the Target 84
Figure 5-9 Empty Text To Send Field 85
Figure 5-10 WaitImage Tab in SendKeys Properties 86
Figure 5-11 Timeout Tab 87
Figure 5-12 Record and Playback Buttons 90
Figure 5-13 Script Menu 91
Figure 5-14 Script Properties Screen 93
Figure 5-15 History Tab in Script Properties 94
Figure 5-16 General Tab in Script Properties 95
Figure 5-17 Dataset Tab in Script Properties 95
Figure 5-18 WAP Browser Test Script 98
Figure 5-19 Saving Script Changes 99
Figure 5-20 Script Status Screen 100
Figure 5-21 Wap Browser Test Results 101
Figure 5-22 Send SMS Script 103
Figure 5-23 Macro Call in Main Script 104
Figure 5-24 Macro Resources for Send SMS 105
Figure 5-25 Empty Macro Resource—LG VX8500 106
Figure 5-26 Reset to Idle—LG VX8500 107
Figure 5-27 Reset to Idle—BlackBerry 8100 Pearl 108
Figure 5-28 Reset to Idle (Macro Instance—BlackBerry 8100 Pearl) 109
Figure 5-29 Empty Macro Instance—Samsung A707 110
Figure 5-30 Reset to Idle (Macro Instance—Samsung A707) 111
Figure 5-31 Send SMS—Samsung A707 Macro Instance (Top) 114
Figure 5-32 Send SMS—Samsung A707 Macro Instance (Bottom) 115
Figure 5-33 Samsung A707 Power key 117
Figure 5-34 Samsung A707 Idle Screen 117
Figure 5-35 Verify Messaging Screen 118
Figure 5-36 Message Sent Screen 120
Figure 5-37 Send SMS Local Variables 122
Figure 5-38 Insert Num Variable in SendKeys 123
Figure 5-39 Properties Screen for the Send SMS Macro Call 124
Figure 5-40 Script Properties—Dataset Tab 127
Figure 5-41 Send Message Script with Variables 129
Figure 5-42 Script Variable—Phone Num 130
Figure 5-43 Send SMS Macro in the Resource Manager View 132
Figure 5-44 Edit Macro Parameters 133
Figure 5-45 Parameter (Variable) Type 134
Figure 5-46 Variable Override 135
Figure 5-47 Test Suite View 136
Figure 5-48 Create a New Test Suite 137
Figure 5-49 TestSuite Configuration 138
Figure 5-50 Configure Script/Macro 139
Figure 5-51 Select or Clear Macro Dialog Box 140
Figure 5-52 Current Script/Macro View 141
Figure 5-53 High-level Report of Test Suite Results 141
Figure 5-54 Schedule View 143
Figure 5-55 Schedule Test Suite Screen 144

Figure 5-56 Create a Device Bundle 145
Figure 5-57 Calendar View 146
Figure 5-58 Detailed Log with Images 147
Figure 5-59 Dashboard View 148
Figure 6-1 *DeviceAnywhere Monitoring* Login 152
Figure 6-2 Monitoring View 153
Figure 6-3 Projects created under Monitor Scripts 155
Figure 6-4 Create a Folder for Monitor Scripts 155
Figure 6-5 Empty Script in the Script Editor 156
Figure 6-6 Add to Script 157
Figure 6-7 Acquire Device Menu 157
Figure 6-8 Scheduler View 159
Figure 6-9 Production Monitor Menu 161
Figure 6-10 Monitor Script and Device Bundles 162
Figure 6-11 Create a Device Bundle 163
Figure 6-12 Monitor Script Execution Frequency 164
Figure 6-13 SLA Policies 165
Figure 6-14 Add New Error Category 167
Figure 6-15 Add new Error Type 168
Figure 6-16 Add New SLA for Success Rate 171
Figure 6-17 Timer Performance SLA 173
Figure 6-18 Video Performance SLA 176
Figure 6-19 Action Path 178
Figure 6-20 Add Escalation Path 179
Figure 6-21 Add Resolution Path 181
Figure 6-22 Dashboard View 182
Figure 6-23 Log from Previous Runs 183
Figure 6-24 Preview Run View 185
Figure 6-25 Video Monitor View 187
Figure 6-26 Add View to the Video Monitor 189
Figure 6-27 Video Monitor View 190
Figure 6-28 Open Violation in Video Player 192
Figure 6-29 Composite View of Video Monitors 192
Figure 6-30 SendKeys Command Dragged to the Target 194
Figure 6-31 Empty Text To Send Field 195
Figure 6-32 Play, Stop, Pause, and Restart Buttons 197
Figure 6-33 Monitor Script Properties Screen 200
Figure 6-34 History Tab in Script Properties 201
Figure 6-35 General Tab in Script Properties 202
Figure 6-36 Dataset Tab in Script Properties 202
Figure 6-37 Send Message Script with Toggle Timers 203
Figure 6-38 Toggle Timer Properties Screen 205
Figure 6-39 Monitor Script Results 206
Figure 6-40 ESPN News Script 208
Figure 6-41 Samsung A707 Cache Cleared 215
Figure 6-42 Start Launch CV Toggle Timer 216
Figure 6-43 Toggle Timers 217
Figure 6-44 Wait Video Properties Screen 218
Figure 6-45 Monitor Script Results 221
Figure 6-46 Monitor Summary 222

Figure 7-1 Checkpoint Tab 223
Figure 7-2 SendKeys Properties Screen 226
Figure 7-3 WaitImage Tab in SendKeys Properties 227
Figure 7-4 Timeout Tab 229
Figure 7-5 Checkpoint Tab in SendKeys Properties 230
Figure 7-6 Hardware Control 231
Figure 7-7 PlayAudio 231
Figure 7-8 FindAndTouch Screen 232
Figure 7-9 Wait Command 233
Figure 7-10 WaitText Command 234
Figure 7-11 WaitImage Properties Screen 236
Figure 7-12 WaitEvent Command 238
Figure 7-13 NavTo Command 240
Figure 7-14 SetVariable Command 241
Figure 7-15 Branch Command 243
Figure 7-16 Variable Expression for Branch1 243
Figure 7-17 Loop Command 244
Figure 7-18 Toggle Timer Properties Screen 245
Figure 7-19 Timers Tab 246
Figure 7-20 VM Command Properties Screen 247
Figure 7-21 Macro Command 248
Figure 7-22 Java Extension 249
Figure 7-23 Success Command 250
Figure 7-24 Fail Command 251
Figure 9-1 Resource Manager View with Libraries 252
Figure 9-2 Idle Screen Resource Instances 254
Figure 9-3 Add Resource Library 256
Figure 9-4 New Folder Added 256
Figure 9-5 Add to Resource Instance 258
Figure 9-6 Resource Instances (Not Implemented) 258
Figure 9-7 Resource Properties 259
Figure 9-8 Macro Script Editor in Resource Manager 262
Figure 9-9 Macro Instance Drop-Down Menu 263
Figure 9-10 Initialize Device for LG VX4500 264
Figure 9-11 Hardware Control Settings to Initialize Device 265
Figure 9-12 Clear Cache Macro 266
Figure 9-13 Insert a Resource from the Properties Screen 268
Figure 9-14 Drag Call Macro into Script 269
Figure 9-15 Macro Dialog Box 270
Figure 9-16 Send an SMS Message (Top) 273
Figure 9-17 Send an SMS Message (Bottom) 274
Figure 10-1 *MyDeviceAnywhere* Home Page 280
Figure 10-2 Subscriptions in My Account 282
Figure 10-3 Choose Subscriptions Screen 283
Figure 10-4 Account Information 284
Figure 10-5 Usage History 284
Figure 10-6 Project List 285
Figure 10-7 Project Details 286
Figure 10-8 Upload Frame Details 287
Figure 10-9 My Applications View 288

Figure 10-10 Test Results 289
Figure 10-11 Create a Reservation 291
Figure 10-12 Device Search Results without Icons 292
Figure 10-13 Motorola K1 Search Results 293
Figure 10-14 Production Monitoring - View Data 293
Figure 10-15 Production Monitoring - View Saved Reports 295
Figure 10-16 System Admin Tab with Packages Page Open 296
Figure 10-17 Device Distribution Screen 297
Figure 10-18 Create a User Group 298
Figure 10-19 Add a New User 299
Figure 10-20 Add Project Screen 300
Figure 10-21 DA Pro License Allocation 301
Figure 10-22 Account Profile 302
Figure 10-23 Detailed Account Usage by User 303

List of Tables

Table 1-1 *DeviceAnywhere* User Guide 17
Table 3-1 Buttons in the Video Control Bar 37
Table 4-1 Set Alarm Script 61
Table 5-1 WAP Browser Test Script 97
Table 5-2 Reset to Idle 112
Table 5-3 Send SMS—Samsung A707 115
Table 6-1 Send Message Script with Toggle Timers 204
Table 6-2 ESPN News Monitoring Script 209
Table 6-3 Clear Cache Macro 213
Table 9-1 Clear Cache Macro 265
Table 9-2 Clear Cache Macro 267
Table 9-3 Send an SMS Message 275

Introduction

Welcome to the *DeviceAnywhere*™ User Manual. This document covers all the features you need to know to use *DeviceAnywhere* effectively. The *DeviceAnywhere* product suite contains three products:

- *DeviceAnywhere* allows you to interact with live *DeviceAnywhere* handsets over the Internet. Your live session can be recorded, viewed, and shared with other users. In addition, you can record simple scripts for repetitive testing. These features constitute the core functionality of *DeviceAnywhere*.
- (Optional) *DeviceAnywhere Pro*™ allows you to automate mobile application testing by creating, executing, and scheduling test scripts. Save sets of commands that step through tests on one or more *DeviceAnywhere* handsets to perform functional or regression testing. You can execute scripts ad hoc, or schedule them to run at your conveniences. Test results are saved and viewable through the *MyDeviceAnywhere* website.
- (Optional) *DeviceAnywhere Monitoring*™ allows you to monitor your deployed application, on real handsets, from the customer's point of view. You can create monitor scripts that are running continuously or set to run on a predetermined schedule to measure performance and availability. A flexible reporting engine allows you to view performance trends, and customer-defineable alerts will notify you of performance violations. *DeviceAnywhere Monitoring* also includes Video Monitoring to monitor video performance, such as frame rate and jitter.

DeviceAnywhere Pro and *DeviceAnywhere Monitoring* are enhancements to the *DeviceAnywhere* core functionality. Licensing and activation of these products is acquired separately.

In order to use *DeviceAnywhere*, *DeviceAnywhere Pro,* and *DeviceAnywhere Monitoring* you must have both a *DeviceAnywhere* user account and a *DeviceAnywhere Pro* or *DeviceAnywhere Monitoring* user license. If you do not have either of these, you should contact the administrator of your account, or you can contact us at DASales@DeviceAnywhere.com.

Contact Information

If you have any questions about how to use any of the *DeviceAnywhere* features, or have an issue to report, please contact us at DASupport@deviceanywhere.com. Click on Help to view online documentation. Click on Report an issue in *DeviceAnywhere Studio* or the *MyDeviceAnywhere* website to send us a problem report.

For questions about additional products invoices, or pricing, please contact DASales@deviceanywhere.com.

1.1 *DeviceAnywhere* System Requirements

While *DeviceAnywhere* can be run on most standard computer configurations, preferred system requirements are as follows:

- 2 GHz Pentium 4 processor
- 1024 MB RAM (1GB RAM), 2GB recommended
- 10 GB Hard Disk Drive available
- Java JRE 1.5 is required, JRE 1/6 is optional (note: during the *DeviceAnywhere Studio* installation process, your system will be checked for this software, and it will automatically be downloaded if required)
- Monitor that is capable of at least a 1024x768 screen resolution
- Broadband Internet connection recommended
- Optional: Audio card for sound input/output

1.2 About This User Guide

This document describes the functionality of *DeviceAnywhere* Version 4.2. It contains descriptive information and many procedures to help you use *DeviceAnywhere* products. This User Guide is divided into chapters that are complete for each feature. You only need to read those chapters that are pertinent to your product licensing. Table 1-1 provides a description of the chapters in this document.

Table 1-1 *DeviceAnywhere* User Guide

Chapter	Description	Recommended Reading
Introduction	This section provides an overview of *DeviceAnywhere* proprietary technology and a list of the hardware needed to support *DeviceAnywhere* installation.	All users
Chapter 1	This section describes the *DeviceAnywhere* User Interface.	All users
Chapter 2	This section describes how to get started by logging in to *DeviceAnywhere* and *DeviceAnywhere Studio.*	All users
Chapter 3	This section describes Device Access, where you interact with live *DeviceAnywhere* handsets, review recorded sessions, and upload test results. You can invite your team members to collaborate in a test session.	All users
Chapter 4	This section describes Record & Playback, a simple scripting feature that allows you to create end-to-end scripts by recording key presses.	All users

Table 1-1 *DeviceAnywhere* User Guide (Continued)

Chapter 5	This section describes *DeviceAnywhere Pro*, a full-featured scripting editor. This feature allows you to create complex test scripts, including variables, macros, and program logic, to perform functional and regression testing. You can create, manage, save, and execute tests and Test Suites across networks and groups of handsets.	This section is reserved for users who purchased a license for *DeviceAnywhere Pro*.
Chapter 6	This section describes *DeviceAnywhere Monitoring*. This feature allows you to create, manage, and execute monitoring scripts to measure network metrics, access video quality and benchmark services in competitor networks.	This section is reserved for users who purchased a license for *DeviceAnywhere Monitoring*.
Chapter 7	This section describes the Resource Manager. This feature allows you to create libraries of resources such as key presses, reference images, and Macros to use during scripting.	This section is for users who purchased a license for *DeviceAnywhere Pro and/or DeviceAnywhere Monitoring*.
Chapter 8	This section is a reference guide to the DeviceAnywhere Script commands.	This section is for users who purchased a license for *DeviceAnywhere Pro and/or DeviceAnywhere Monitoring*.
Chapter 9	This section describes the *MyDeviceAnywhere* website. This area allows you to manage your projects, review test results, and share information with team members through Email.	All users

Recommended Reading

- When you are only interested in the core functionality of *DeviceAnywhere*, read the Introduction, chapters 1 through 4, and chapter 9. These chapters cover the core functionality and use of the *MyDeviceAnywhere* website.
- When you are interested in Automation with *DeviceAnywhere Pro*, read the Introduction, chapters 1 though 5, and chapters 7 through 9. These chapters cover all the basics and describe how to automate your test scripts for functional and regression testing across sets of devices.
- When you are interested primarily in *DeviceAnywhere Monitoring*, read the Introduction, chapters 1 though 4, and chapters 6 though 9. These chapters cover the basics and describe how to create and schedule scripts that measure your mobile applications' performance or help you benchmark services in your mobile products.

Documentation Resources

The following *DeviceAnywhere* documents are available:

- *DeviceAnywhere* User Guide Version 4.2
- *DeviceAnywhere* Quick Start Guide 1.1
- *DeviceAnywhere* Troubleshooting Guide Version 1.1
- *DeviceAnywhere Pro* Scripting Tutorial 1.1
- *DeviceAnywhere Pro* Best Practices 1.0
- *DeviceAnywhere* Release Notes Version 4.2

To access documents online, click Help in *DeviceAnywhere Studio* or in the *MyDeviceAnywhere* website.

Typographical Conventions

This document uses the following typographical conventions. An example illustrates each convention.

Convention	Meaning or Use	Example
Italic font	References to Mobile Complete products	In order to use *DeviceAnywhere Monitoring*, you must have both a *DeviceAnywhere* user account and a *DeviceAnywhere Monitoring* user license.
Underlined Text	Indicates a link to a web page	http://www.deviceanywhere.com
Single quote (')	Indicates special terms used in *DeviceAnywhere*	This lesson also introduces the all-important concept of 'checkpoints,'...
Blue Text	Cross-reference link to a location in the current document	Figure 2-1 illustrates the *MyDeviceAnywhere* home page.

1 DeviceAnywhere Studio Interface

Once you are logged in to *DeviceAnywhere Studio*, the Device Access view is displayed as the default view. Other major features and views of *DeviceAnywhere Studio* are accessible through the Icon Bar on the far left of the screen. The Device Access view illustrates the basic layout of all views in *DeviceAnywhere Studio*.

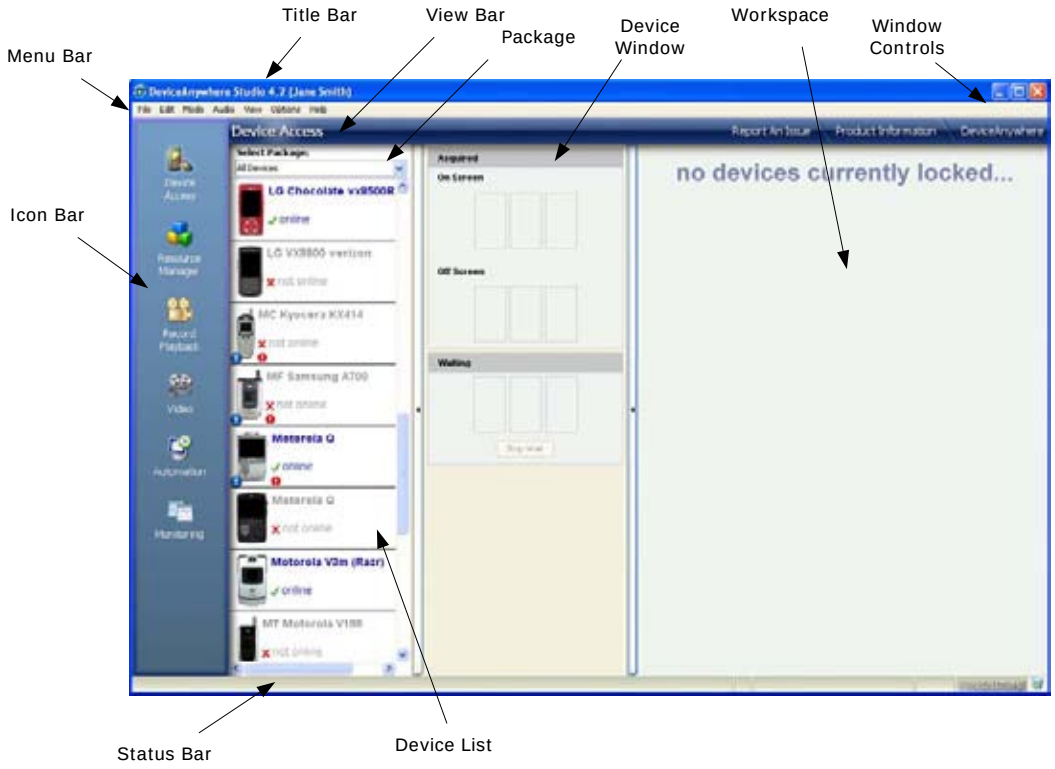


Figure 1-1 Device Access View

Figure 1-1 illustrates the components of the *DeviceAnywhere Studio* interface.

- Title Bar—The Title Bar shows the *DeviceAnywhere Studio* product name and version of the software. The name of the registered user is shown in parentheses.
- Menu Bar—The Menu Bar provides a set of drop-down menus for each command in *DeviceAnywhere Studio*. The commands in the Menu Bar change based on the current view. In the Device Access view, the following commands are included: File, Edit, Mode, Audio, View, Options, and Help.
- View Bar—The View Bar shows the current working view in *DeviceAnywhere Studio*. It provides two links: 1) Report an Issue is used to send a problem report to *DeviceAnywhere* customer support; 2) *MyDeviceAnywhere* is a link to navigate to the *MyDeviceAnywhere* home page.
- Package—This window shows the current package that is assigned to your project. A package is a set of handsets.
- Device Window—This window displays the devices you are working with during an interactive session only. It includes devices in three states: 1) devices displayed in the workspace, 2) devices placed off the screen, and 3) devices that you have reserved.
- Window Controls—These are standard controls that allow you to maximize or minimize the window.
- Icon Bar—The Icon Bar in the left panel of the screen allows you to switch to other views in *DeviceAnywhere Studio*. It launches special product features such as *DeviceAnywhere Pro*™ or *DeviceAnywhere Monitoring*™. These features are optional and are based on separate licensing. When you click an icon, the icon flashes to indicate your choice of views. The Icon Bar is available in all views of *DeviceAnywhere Studio*.
- Explorer (not shown here)—The Explorer provides access to features, folders, scripts, and macros, depending on the current view. Except for the Device Access view, the Explorer appears above the Device list.
- Workspace—The large right panel of the screen is the primary workspace for all views. In this space, you see phones acquired, test scripts under development, or even test reports. In Device Access view, you can step through recorded frames from a live session with Device Video Controls.
- Device List—The Device List (below the Explorer) shows the status of all handsets in the package. A green check indicates that a device is available. A red circle shows the device is occupied by another user. A red X shows the device is out-of-order. The contents of the panel below the Explorer always changes, based on the context of the current view.
- Status Bar—The Status Bar at the bottom of the screen provides system messages and error feedback.

1.1 Command Entry

DeviceAnywhere Studio provides a flexible interface that allows you to enter commands through the Menu Bar.

The Menu Bar contains the following menus:

- File—The File menu allows you to navigate directly to your MyDeviceAnywhere account. The Enter code option enables you to share a device with another user by entering the Shared device code and pressing Send. Select Exit from the drop-down menu when you want to exit DeviceAnywhere Studio.
- Edit—The Edit menu supports copy and paste functions. *DeviceAnywhere Studio* also supports standard keyboard shortcuts, such as Control+C and Control+V.

- Mode—The Mode menu allows you to set the Key mode for an entry in the device. From this menu, you can select Alpha, Numeric, or URL modes.
- Audio—The Audio menu allows you to enable or disable the speaker and to change the size of the audio buffer.
- View—The View menu allows you to switch views in *DeviceAnywhere Studio*. It has the same function as the commands on the left Icon Bar.
- Options—The Options menu provides access to Error Management features that support DeviceAnywhere Monitoring. The Settings option enables you to set three settings: 1) Ad Hoc Video SLA Wait Time, 2) Maximum number of devices on screen, and 3) Wake up warnings. It also includes the About tab for a brief description of DeviceAnywhere Studio.
- Help—The Help menu provides access to the latest DeviceAnywhere documentation. It also supports many other links as described below.

1.2 *DeviceAnywhere* Help

DeviceAnywhere Studio supports online documentation and Help for all features of the product. The Help link is available for quick access to information while you work. Press F1 on your keyboard to view the Help system.

The Help link provides access to the following:

- Help—This link provides access to the online Help.
- Open User Manual—This option opens the current DeviceAnywhere User Manual in a separate window.
- Open Troubleshooting Guide—This option opens the DeviceAnywhere Troubleshooting Guide. The Troubleshooting Guide provides a comprehensive description of issues you may encounter while uploading the DeviceAnywhere Studio. DeviceAnywhere is easy to load if you know how the process works.
- Report an Issue—This option allows users to report issues to the DeviceAnywhere support team.
- Info Pop-ups on—This option turns information pop-ups on or off. When Info Pop-ups is enabled, and you select Preview Run in DeviceAnywhere Monitoring, a popup appears on completion of the monitoring script. Clicking on the popup directs you to Production Monitoring in your MyDeviceAnywhere account to view the results.
- About—This displays the *DeviceAnywhere* copyright notice, version number and other information about your installation of *DeviceAnywhere*.

1.3 *DeviceAnywhere* Unexpected Termination

When DeviceAnywhere terminates unexpectedly, the active devices are retained for five minutes. When DeviceAnywhere is re-launched, the screen shows the acquired devices from the previous session and enables you to select how a device is handled after the re-launch.

Figure 1-2 shows the Recovered Devices screen.

Figure 1-2 Recovered Devices

How to Recover the Active Devices after an Unexpected Termination:

1. Select retain or release for each device in the previous session.
2. Click Continue.

DeviceAnywhere resumes with the devices that you have selected.

2 Getting Started

2.1 Log in to *DeviceAnywhere*

To start using *DeviceAnywhere Studio*, first log in to *DeviceAnywhere*.

- Launch your web browser.
- Enter the URL: http://www.deviceanywhere.com.
- Enter your username and password in the upper right corner of the screen.

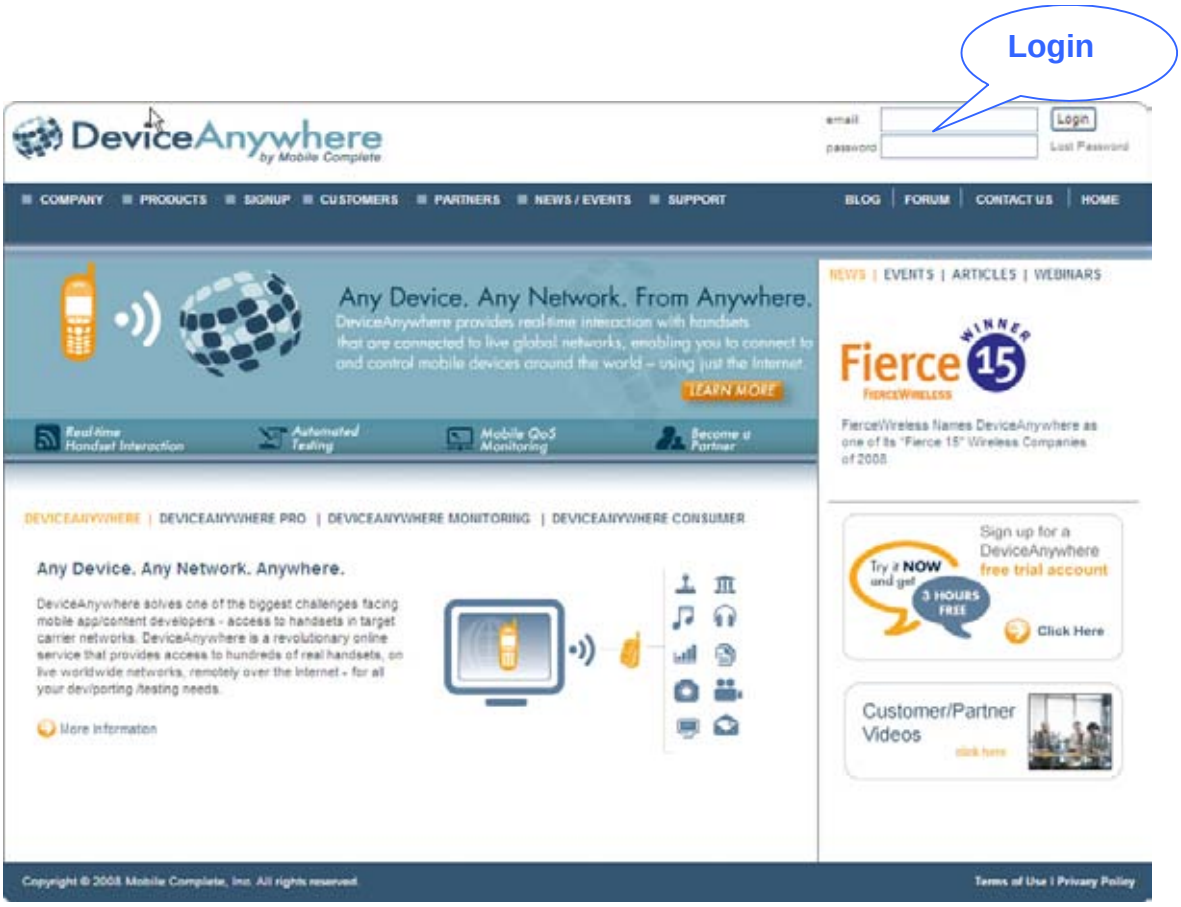

Login
DeviceAnywhere
by Mobile Complete
email
password
Login
Lost Password
COMPANY
PRODUCTS
SIGNUP
CUSTOMERS
PARTNERS
NEWS / EVENTS
SUPPORT
BLOG
FORUM
CONTACT US
HOME
Any Device. Any Network. From Anywhere.
DeviceAnywhere provides real-time interaction with handsets that are connected to live global networks, enabling you to connect to and control mobile devices around the world – using just the Internet.
LEARN MORE
Real-time Handset Interaction
Automated Testing
Mobile QoS Monitoring
Become a Partner
NEWS | EVENTS | ARTICLES | WEBINARS
WINNER
Fierce 15
FierceWireless Names DeviceAnywhere as one of its "Fierce 15" Wireless Companies of 2008
DEVICEANYWHERE | DEVICEANYWHERE PRO | DEVICEANYWHERE MONITORING | DEVICEANYWHERE CONSUMER
Any Device. Any Network. Anywhere.
DeviceAnywhere solves one of the biggest challenges facing mobile app/content developers - access to handsets in target carrier networks. DeviceAnywhere is a revolutionary online service that provides access to hundreds of real handsets, on live worldwide networks, remotely over the Internet - for all your dev/porting /testing needs.
More Information
Sign up for a DeviceAnywhere free trial account
Try it NOW and get 3 HOURS FREE
Click Here
Customer/Partner Videos
Copyright © 2008 Mobile Complete, Inc. All rights reserved.
Terms of Use | Privacy Policy

Figure 2-1 *DeviceAnywhere* Login

Note: Log in by using the username and password provided by your administrator. If you don't have a username and password, contact your system Admin.

2.2 *MyDeviceAnywhere* Website

When you log in to *DeviceAnywhere*, the *MyDeviceAnywhere* home page appears on your screen. The *MyDeviceAnywhere* website allows you to manage your projects and view test results. The website provides settings to administer your account.

Figure 2-2 illustrates the *MyDeviceAnywhere* home page.

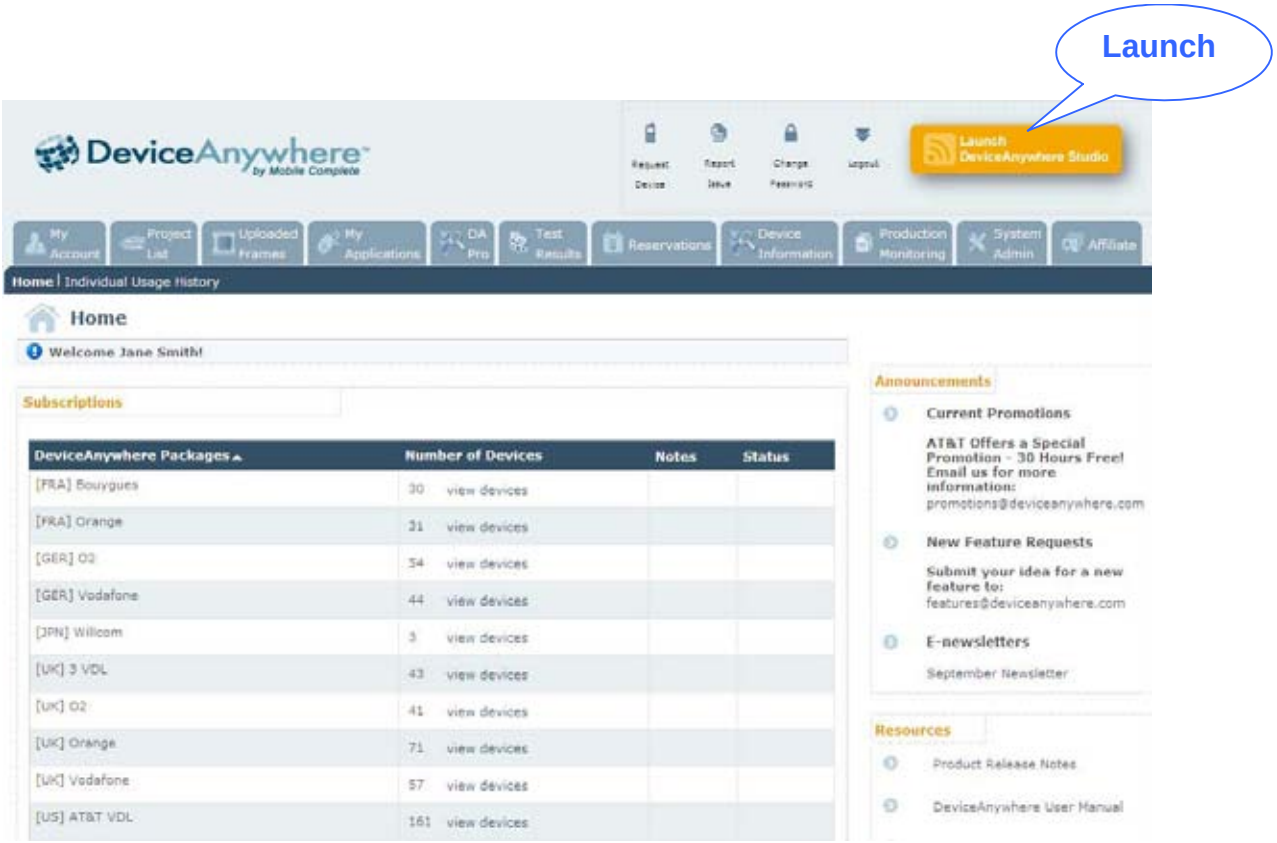


Figure 2-2 *MyDeviceAnywhere* Home Page

2.3 Launch *DeviceAnywhere Studio*

Click Launch *DeviceAnywhere Studio* in the upper right corner of the screen. When you launch *DeviceAnywhere Studio*, your system is checked for the required version of the Java Runtime Environment (JRE). If the required version is already installed on your system, *DeviceAnywhere Studio* begins to download. If the necessary version is not found, JRE is automatically installed as part of the installation process.

The initial launch of *DeviceAnywhere Studio* takes up to 5 minutes depending on your broadband or telephone connection. Installation takes longer, since the system is downloading the device information for the first time.

As soon as the download is complete, you are asked if you want to create desktop shortcuts for *DeviceAnywhere Studio*. Select Yes to create *DeviceAnywhere Studio* icons on your desktop and

in the Start menu. In the future, you can launch *DeviceAnywhere Studio* by double clicking on these shortcuts.

> Note: Depending on your Web browser security settings, you may see a Security Warning asking whether you trust this content from Mobile Complete, Inc. Please click Yes or Always when prompted.

2.4 Log in to *DeviceAnywhere Studio*

After launching *DeviceAnywhere Studio*, you see another login screen. Enter again the username and password given to you by your administrator.

When you log in successfully to *DeviceAnywhere Studio*, the Device Access view appears on your screen.

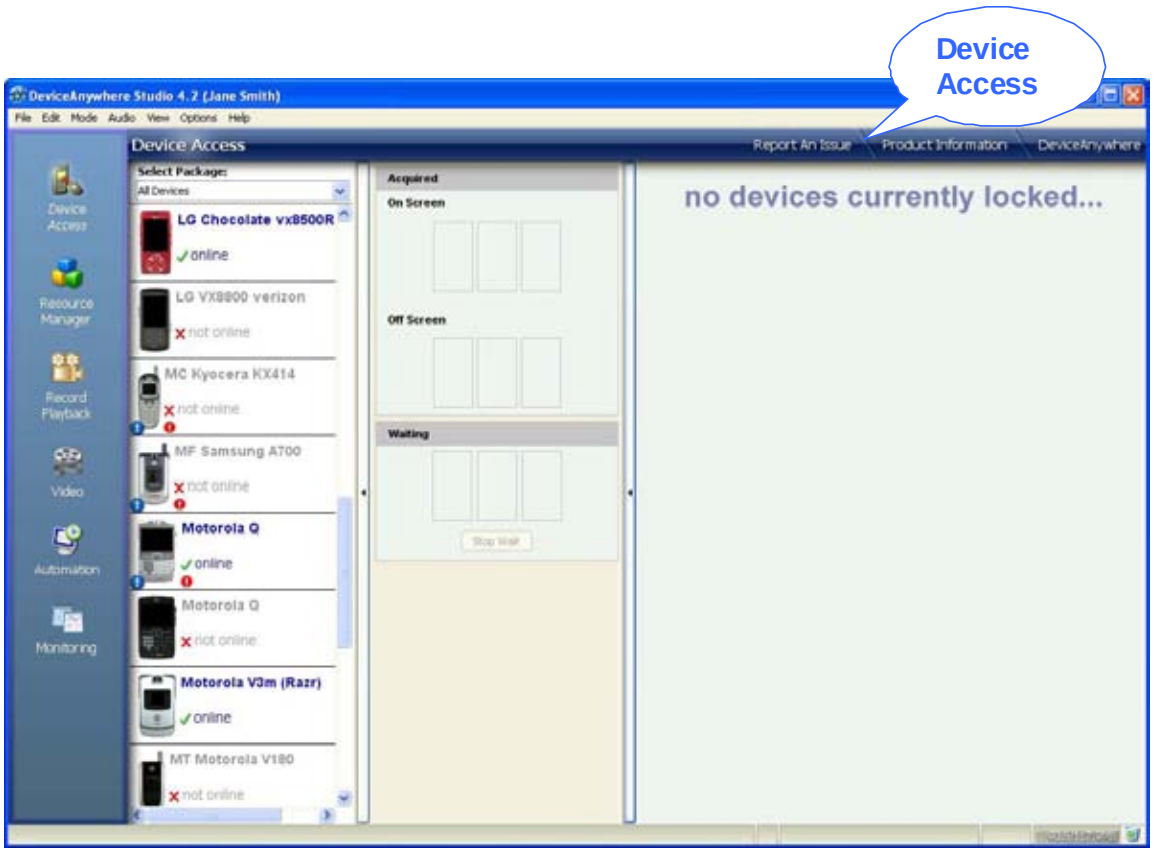


Figure 2-3 Device Access View

3 Device Access

DeviceAnywhere is designed to support an optimal testing environment for wireless handsets and applications. *DeviceAnywhere* core proprietary technology enables remote (over the Internet) access to real handsets in live carrier networks.

Additionally, collaboration features built into *DeviceAnywhere* allow for quick capture and replay of on-device activity as frame-by-frame video and movie clips. *DeviceAnywhere* automatically records the live session from the first key press, which can be viewed and uploaded to the *MyDeviceAnywhere* website. This feature enhances the interactions among distributed teams and shortens development time.

DeviceAnywhere enables developers to perform in-network, on-device testing of mobile handsets and applications from the convenience of their desktops. They have remote access to all aspects of handsets, including the buttons, LCD screen, ringer, speaker, battery, power-plug, data-cable, and so on.

3.1 *DeviceAnywhere* System Setup

To get the best performance from DeviceAnywhere Studio, two adjustments should be taken into consideration. These adjustments may be needed only the first time you launch DeviceAnywhere Studio on your system:

- Change the resolution of your monitor to 1280×1024 (recommended)
- Set the color depth of the monitor

3.1.1 Change the Resolution of your Monitor

For information to change the resolution of your monitor, refer to the User Guide that came with your equipment. You can also refer to your operating system's online Help.

How to Change Monitor Resolution in Microsoft Windows:

1. Click the Start menu.
2. Select the Control Panel icon.
3. Double click the Display icon to open the Display screen.
4. Select the Settings tab.
5. In the Screen Resolution window, move the slider bar to adjust the resolution to at least 1280 by 1024 pixels.
6. Click OK.

The monitor adjusts screen resolution and prompts to save changes.

3.1.2 Set the Color Depth of the Monitor

DeviceAnywhere Studio enables you to select the number of colors you want to view on the device screen.

How to Set the Color Depth:

1. Right click on a device in the Device list.
2. Select Access Device.
3. Select Acquire Device.
4. Right click on the device in the workspace. The Device menu appears.
5. Select Color Depth.
6. Select from the options available: 1) 65536 colors, 2) 4096 colors, or 3) 128 colors.

Note: Over a slow bandwidth connection, you may find it useful to reduce the number of colors to improve response time.

3.2 Device Access Overview

Three windows support the primary features of interactive device testing:

- Package Window—The Package Window displays the current package selected. This setting depends on the packages assigned to your project by your administrator.
- Device Window—This window shows the status of devices: 1) Acquired devices in the workspace, 2) Acquired devices that have been placed off screen, and 3) Devices that are on a reserved list.
- Device List—The Device list shows all the devices that can be used in testing based on availability.
- Workspace—The workspace enables you to see a device, press keys, and play back the frames recorded during a live session.

Figure 3-1 shows the components of the Device Access view.

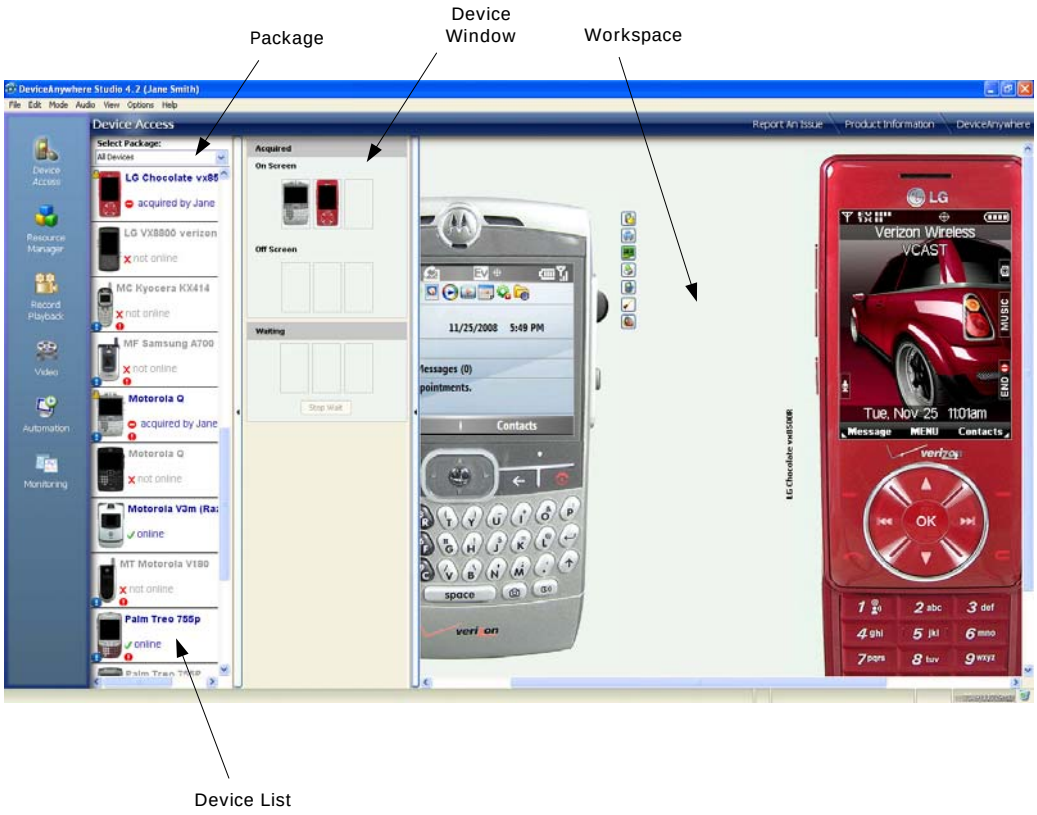


Figure 3-1 Device Access View

For a complete description of the *DeviceAnywhere* Interface, refer to "DeviceAnywhere Studio Interface" on page 20 of this document.

3.2.1 Device Menu

The Device menu in the Device Access view enables you to control the device and enter important settings during a session.

To Display the Device menu, do the following:

- Right clicking on a device in the Device list displays a short Device menu. The Device menu lets you access and acquire a device.
- Right clicking on a device in the workspace displays the full Device menu. This Device menu provides a full set of commands to perform operations with the device.

Figure 3-2 shows the full Device menu opened in the workspace.

Figure 3-2 Device Menu

The Device menu contains the following options:

- Access Device—This option enables you to Reset a device or Release a device when you are done with it.
- Share Device—This option enables you to invite another user to view the device screen for the device that you are working with during a live session. Three methods for inviting a user are available. You can generate a code, send an email, or select users from an online list.
- Color Depth—This option colors the number of colors viewed on your computer screen.
- Hardware Control—This option controls the hardware functions on a device such as Disconnect battery, Attach data cable, or close flip phone.
- Audio—This option controls the size of the audio input and output buffers.
- BREW—This option allows you to control functions relating to uploading, downloading, and managing BREW applications on a device.
- Export—This options enables you to generate frames and video clips, which can be saved to MyDeviceAnywhere.
- Paste from Clipboard—This option allows you to paste data (Alpha, numeric, or URL) into an entry field on a device.
- Reserve Device—This option allows you to reserve time for a particular device based on the schedule that you specify.
- Report an Issue—This option enables you to send an email to DeviceAnywhere in order to report an issue.
- Run Cleanup Scripts—This option enables you to run cleanup scripts that initialize a device and restore it to default settings.
- Properties—This option opens the Device Properties screen, where important information about the device is provided.

When a device supports BREW applications, a BREW option appears in the Device menu as well.

3.2.2 On-Screen Icons

When a device is acquired, icons are shown in the top right corner of the device. These icons execute the same commands that are found in the Device menu.

The following commands are available on-screen:

- Access Device
- Share Device
- Hardware Control
- Export
- Properties
- Run Cleanup Scripts
- Press to Enable Audio

Figure 3-3 shows the on-screen icons in the top right corner of the device.

Figure 3-3 On-Screen Icons

3.3 Device Session

The Device Access view enables you to interact with devices in real-time. This section describes the commands and procedures to conduct a session.

3.3.1 Access a Device

How to Start an Interactive Session:

1. Right click a device in the Device list. The Device menu appears.
2. Select Access Device.
3. Select Acquire Device.

As soon as you acquire the device, DeviceAnywhere studio goes into interactive mode. Notice that the device is active and appears in the workspace. When you acquire a device, you prevent other users from accessing it until you release it. Now the device is ready to record key presses.

When more than one device is acquired, the devices are added to the workspace. Use the slider bar to pan across the workspace. You can work with an unlimited number of devices in the Device Access view, limited generally to your Package subscriptions. You are charged for each device that you acquire.

3.3.2 Device Window

The Device window provides a visual confirmation of the devices that you are managing.

You can also control the number of devices displayed in the Device window by changing the setting through the Options menu in the Menu Bar.

How to Set the Number of Devices in the Device Window:

1. Select Options in the Menu Bar.
2. Select Settings.
3. Under Acquired Devices, set the maximum number of devices on screen by pressing on the drop-down menu. Five devices is the maximum allowed.

Refer to Figure 3-1. This figure illustrates the workspace with two devices and the slider bar (below the workspace) that enables you to move from one device to the next.

3.3.3 Device Availability and Status

All test devices associated to your account are accessible through the Package window and Device list.

- Package window—This window shows the current Package selected. Make sure to select the correct package for the project you are working on. The Packages shown in the window correspond to the subscriptions for your account.
- Device List—The Device list shows all the devices associated to a Package. Device availability and status is shown via icons next to each device.

Device Availability and Status Icons are given as follows:

- Green checks indicate available devices.
- Red circles indicate devices that are in use.
- Yellow locks show a device in use.
- Blue circles indicate a note is entered in the Status tab.
- Red Xs indicate devices that are out-of-order.

3.3.4 Wait for a Device

In case a device is acquired by another user, you can select the Wait for Device option in the Device menu. Once the previous user releases the device, the device is automatically acquired by the first person in the queue.

How to Wait for a Device:

1. Right click on the device. The Device menu is displayed.
2. Select Access Device.
3. Select Wait for Device.

3.3.5 Release a Device

If you already have acquired a device, right clicking on the device displays the drop-down menu. Select the Release Device option to release the current device.

How to Release a Device:

1. Right click on the device. The Device menu is displayed.
2. Select Access Device.
3. Select Release Device.

3.3.6 Reset Device

When you select the Reset Device option, the battery for the device is disconnected and reconnected. This option rapidly resets the device.

3.3.7 Wake Up a Device

DeviceAnywhere supports a feature that enables you to execute a Wake up Device option from the Device Access on-screen icon. A device with Wake up Device enabled is highlighted in gray when the device begins to enter Sleep mode. When the device finally shuts down, the device screen goes blank.

The procedure below describes the steps needed to wake up the device.

How to Wake Up a Device:

1. Press any key on the keyboard, except the Power key. The Wake up Device prompt screen is displayed.
2. Press on the Device Access on-screen icon. A drop-down menu is displayed.
3. Select Wake Up Device. The device is restarted within seconds.

Figure 3-4 shows the prompt screen that appears when you press any key on the device after the device has entered Sleep mode.

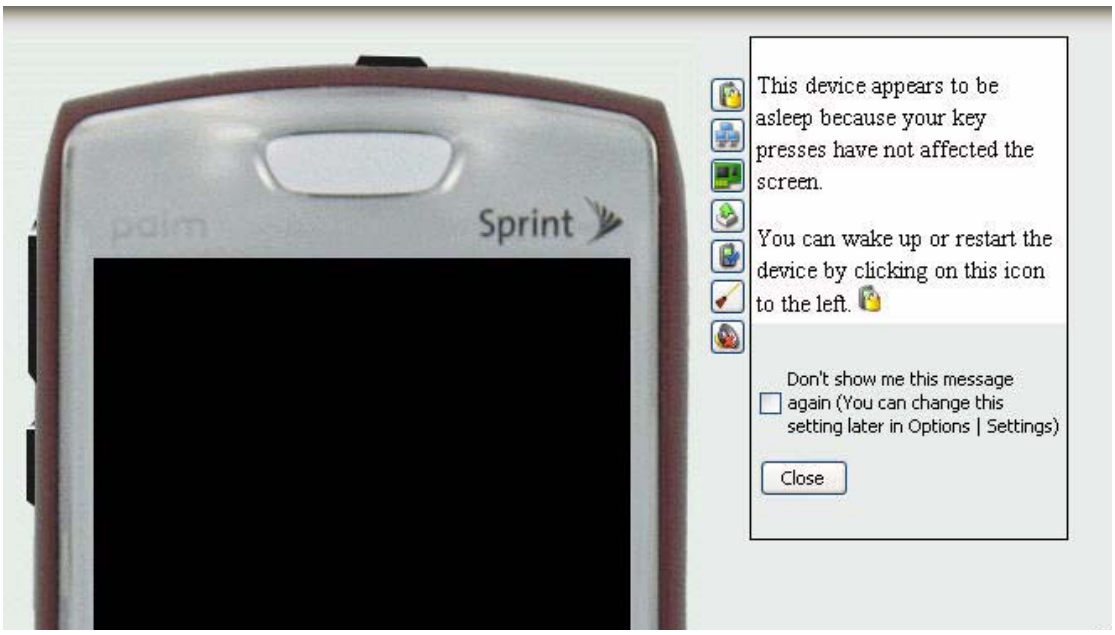

palm
Sprint
This device appears to be asleep because your key presses have not affected the screen.
You can wake up or restart the device by clicking on this icon to the left.
Don't show me this message again (You can change this setting later in Options | Settings)
Close

Figure 3-4 Wake Up Device Prompt Screen

This figure illustrates the Wake Up Device option enabled in the drop-down menu.

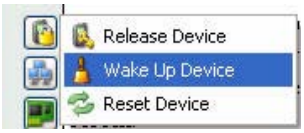


Figure 3-5 Wake Up Device

3.3.8 Run Cleanup Scripts

When you have finished working with a device and release it, DeviceAnywhere automatically enters a device reset and cleanup cycle. Device reset and cleanup is implemented by running test scripts that execute commands on the device to restore the device to the default settings and wipe the device clean by clearing the cache, deleting the Inbox, and so on.

DeviceAnywhere reset and cleanup scripts are now available through the Device menu. Right click a device in the workspace and select Run Cleanup Scripts. This action resets the device to the default settings and wipes the device clean.

If you are loading proprietary applications on a DeviceAnywhere handset, make sure that the application is deleted from the device before you release it. Ensure that your proprietary applications are safeguarded against unintended distribution.

3.4 Device Properties

The Device Properties screen displays different property values for the device.

Five tabs provide device information as described below:

- Device—The Device tab shows the location, Ensemble server where the device is hosted, the phone number of the device, device serial number and MCD. You often need to look up the phone number of a device during testing. Always look in the Device Properties tab for this information.
- Device Type—The Device Type tab shows the manufacturer's model number for the phone and a brief description of its features.
- SIM—The SIM tab shows the SIM number for the device used to identify the device on a carrier network.
- Features—The Features tab is a view-only tab for general users. On this tab, administrators enter features of the device.
- Status—The Status tab enables you to enter notes on the status of the device.

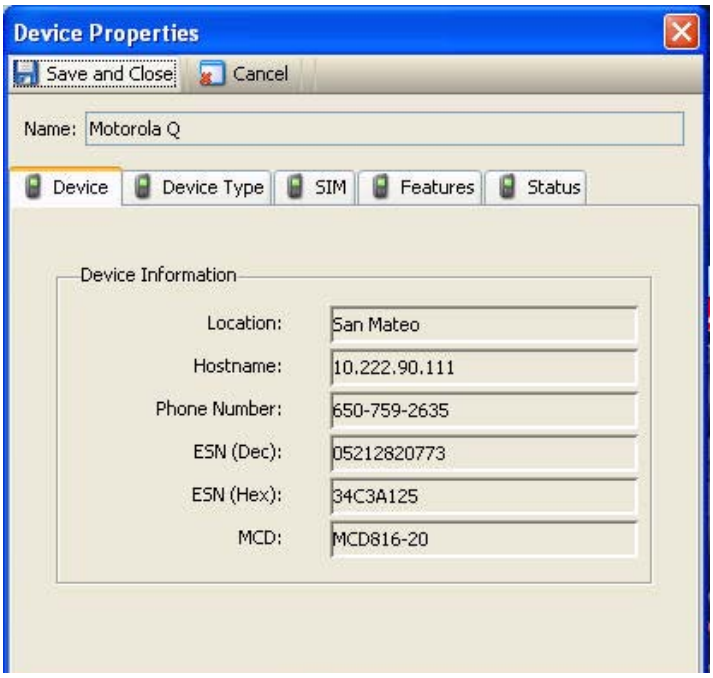


Figure 3-6 Device Properties

3.5 Device Video Controls

Figure 3-7 shows an acquired device with the Video Control Bar below the device.

Figure 3-7 Acquired Device with Video Control Bar

DeviceAnywhere Studio records up to 15 minutes of device activity. You can review this activity by using the Video Control Bar displayed below the device.

Any frame presented on the device's Liquid Crystal Display (LCD) is recorded as a part of the device session. When the device is acquired, you can navigate between the captured frames using the Video Control Bar.

The table below describes the buttons in the Video Control Bar.

Table 3-1 Buttons in the Video Control Bar

	Scan Backward—This button scans backward by many frames.
	Move Backward—This button moves the recording backward by one frame.
	Move Forward—This button moves the recording forward by one frame.
	Scan Forward—This button scans forward by many frames.
	Pause—This button pauses the recording.
	Show Live Video—This button resets the device image to show live video.

3.6 Interactive Session

Typically in a session, you acquire a device, upload an application, and begin pressing keys on the device.

3.6.1 Start an Interactive Session

The following is a simple example of an interactive session. In this example, you acquire a device, send a text message, and reset the device to a known state.

The steps outlined below are general, since each device within a network has its own series of key presses to send a text message. You'll have to work out the series of key presses needed for the particular device you are using.

To Start an Interactive Session, follow the procedure below:

1. Choose a device from the Device list. (Look for devices with a green check mark.)
2. Right click on the device to display the Device menu.
3. Select Access Device.
4. Select Acquire Device. The device is active and immediately displayed in the work-space.
5. Press the menu key on the device.
6. Navigate to the New Message (or Send Message) option on the menu by pressing the up, down, left, right arrow keys on the device.
7. Select New Message. The New Message screen appears on the device.
8. Enter the phone number where the text message will be sent.
9. Click to confirm the phone number. The message body screen appears.
10. Enter the message body text: This is a SMS test.
11. Click to confirm the message body text.
12. Click Send. After several seconds, the Message Sent screen appears.
13. Press the End key three times. The device returns to the idle screen.

The interactive session was automatically recorded. You can now review the test results by pressing the buttons on the Video Control Bar. Return to live interaction by pressing Show Live Video, the far right button on the Video Control Bar.

3.7 Key Press Functions

During a live session, you often need to enter text or numeric data into a device. *DeviceAnywhere Studio* supports simple copy and paste commands to enter data. In addition, it enables you to enter multiple key mappings when more than one key is pressed at the same time.

3.7.1 Key Modes

Mobile devices require a key mode when entering data.

DeviceAnywhere supports three Key Modes for data entry:

- Alpha for text
- Numeric for numbers
- Web for URL entry

3.7.2 Copy and Paste Commands

You can paste into a text or numeric entry field in a device (e.g. messaging text, URL address, and so on) by using Paste from Clipboard in the Device menu.

To Use Copy and Paste commands, follow the steps below:

1. First, copy the required text by using the keyboard shortcut Control+C.
2. Then go to the screen on the device where you want to enter the text.
3. Select Paste From Clipboard.
4. Make sure to choose the correct Key mode.
5. The text is then automatically entered into the device.

3.7.3 Holding Down a Key on a Device

Often you need to hold down a key on a device. This action is accomplished by holding down the Shift key on your keyboard, and then pressing the key on the device. This can be useful to turn the power on or off for a device.

3.7.4 Multiple Key Mapping

Some devices support Multiple Key Mapping. This feature enables you to program a function key, such as F5, to correspond to simultaneous device key presses. This feature is supported by custom hardware wiring for select devices.

Multiple Key Mapping must be enabled to operate within your environment. Contact Support@deviceanywhere.com for information on enabling Multiple Key Mapping.

How to Create a Multiple Key Mapping:

1. Navigate to the Multiple Key Mapping screen on the device.
2. Select a function key from the drop-down menu. (F6 is selected below.)
3. Then, click on the keys on the device that you want to map to the function key.
4. Click Update to store the Multiple Key Mapping.

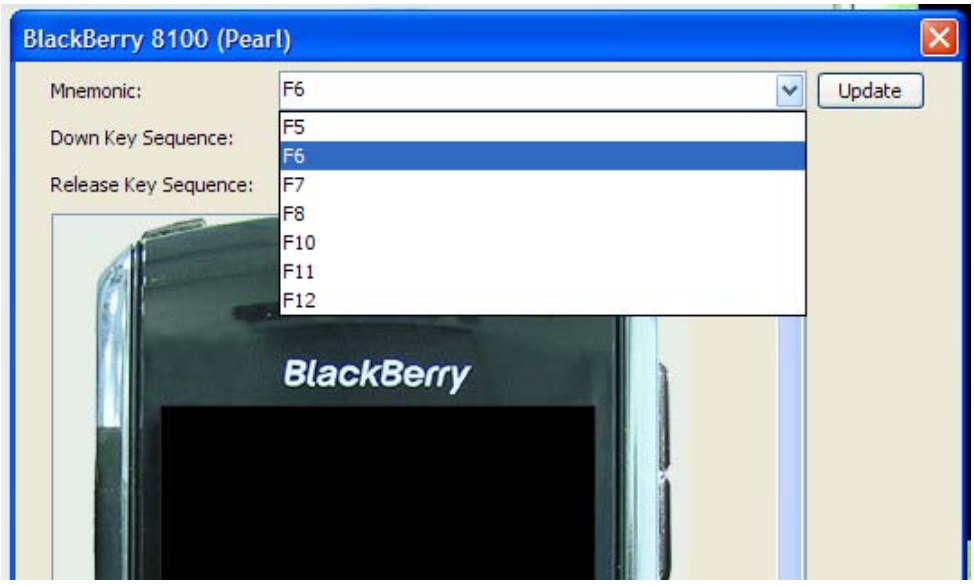


Figure 3-8 Multiple Key Mapping Screen

3.8 Device Audio Controls

The Audio options on the Device menu allow you to control the speaker, the on/off switch for recording, and the audio buffer settings of the device. For example, during a live session, you can record your own voice and play it back over the speaker. Or you could download a ringtone and hear the ringtone over the speaker. These commands support audio feedback during testing.

Depending on the device, Audio options include the following:

- Speaker—The Speaker option enables you to enable and disable the speaker so that you can talk to a person during testing.
- Recording—The Recording option enables you to turn recording on or off through the microphone during a phone call.
- Options—Audio Options enables you to adjust output and input audio buffer settings. For example, increasing the size of the output audio buffer improves audio quality by mitigating delays due to Internet latency.

Figure 3-9 shows the Buffer Properties screen.

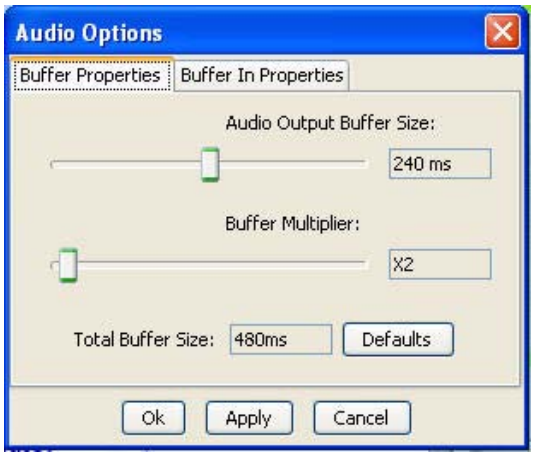


Figure 3-9 Buffer Properties Screen

How to Set the Audio Buffer Size:

1. Right click the device to display the Device menu.
2. Select Audio on the Device menu.
3. Select Options. The Buffer Properties screen is displayed.
4. Adjust the slider under Audio Output Buffer Size to the size you want.
5. Click Apply.
6. Click OK.

The Audio Output Buffer Size is adjusted to the size you selected.

3.9 Hardware Control

Hardware Control lets you control hardware features of the device that are not controllable by the software. This drop-down menu is context sensitive and displays only the hardware controls pertinent to the acquired device.

Depending on the device, Hardware Controls include the following:

- Disconnect Power—Enables you to connect/disconnect the device power cable.
- Disconnect Battery—Enables you to connect/disconnect the battery.
- Turn On Camera Light—Enables you to turn on the camera light.
- Connect Data Cable—Enables you to connect/disconnect the data cable.
- Open Flip Phone—Enables you to open and close a clam shell-style phone, or slide a candy-bar phone.

3.10 Device Collaboration

During a live session, you can collaborate with users. *DeviceAnywhere* enables you to share a device, invite other users, and monitor a live session.

3.10.1 Sharing a Device

With the Share Device option in the Device menu, *DeviceAnywhere Studio* enables other users to view the device(s) you are currently using through their own computer.

How to Share a Device:

1. Right click on the device in the workspace to display the Device menu.
2. Select Share Device. The Share Device dialog box appears.

Alternatively, press the Share Device on-screen icon (three monitors) in the top right corner of the device to execute this command.

Figure 3-10 shows the on-screen icon for Share Device.

Figure 3-10 On-Screen Icon for Share Device

3.10.2 Inviting Users

Figure 3-11 shows the Share Device dialog box. From this dialog box you have three options to invite users to a live session. You can share a device with more than one user at a time.

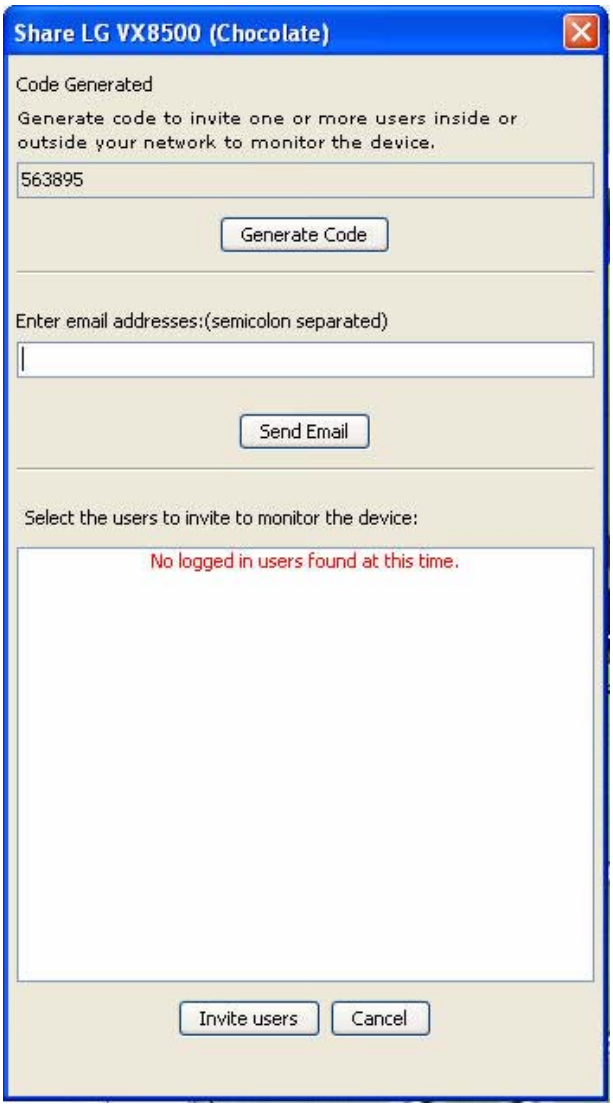

Share LG VX8500 (Chocolate)
Code Generated
Generate code to invite one or more users inside or outside your network to monitor the device.
563895
Generate Code
Enter email addresses:(semicolon separated)
Send Email
Select the users to invite to monitor the device:
No logged in users found at this time.
Invite users
Cancel

Figure 3-11 Share Device Dialog Box

How To Share a Device by Generating a Code:

1. Right click an active device in the workspace. The Device menu appears.
2. Select Share Device. The Share Device dialog box appears.
3. In the top window, click Generate Code.
4. Give or send the code to other users.

Selecting Generate Code generates a numerical code that you can verbally share with other users, or manually email to them. You can invite as many as 25 users to a live session.

How to Send an Invitation to Share a Device by Email:

1. Right click an active device in the workspace. The Device menu appears.
2. Select Share Device. The Share Device dialog box appears.
3. In the Email window, enter the email addresses of team members you'd like to invite to a live session.
4. Click Send Email.

Entering email addresses and selecting Send Email automatically sends an email with a code and instructions on how to view the shared device. The user receiving the email should follow the instructions in the email. Follow the same steps described in "How to Observe a Device using a Code" below.

How to Send a Pop-Up Invitation to Share a Device:

1. Right click an active device in the workspace. The Device menu appears.
2. Select Share Device. The Share Device dialog box appears.
3. In the online user list (bottom window), select the checkbox for team members you'd like to invite to a live session.
4. Click Invite Users. A pop-up invitation is sent to all the users selected.

Note that only users who are currently logged in to *DeviceAnywhere* appear in the online user list. The user receiving the pop-up invitation receives a code and instructions to join the live session. Follow the same steps described in "How to Observe a Device using a Code" below.

3.10.3 Accepting an Invitation

When you are invited to monitor devices, you can accept or reject any invitation you receive. When you accept an invitation, you can observe (monitor) the device.

How to Observe a Device using a Code:

1. In the Device Access view, right click File in the Menu Bar.
2. Select Enter Code. The Code Entry screen is displayed.
3. Enter the code you received when invited to a session.
4. Click Send.

The device for the Shared session immediately appears in the Device window. This shared device takes precedence over any other devices that might be active in the Device window. In this mode, you can only view activity on the shared device.

3.10.4 Stop Observing a Device

How to Stop Observing a Device:

Anytime after accepting an invitation, as an observer you can click the Stop Monitoring button on the toolbar and leave a sharing session.

Figure 3-12 shows the Stop Monitoring button on the top right of the screen.

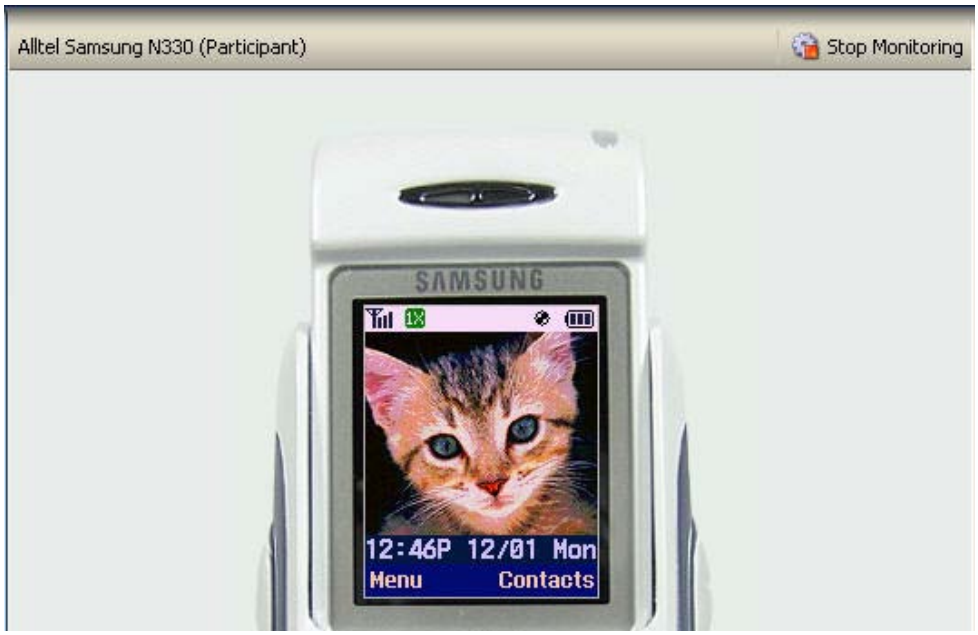


Figure 3-12 Stop Monitoring (Top right)

3.10.5 Sending Messages

While you are sharing devices with other users, you can exchange messages with these users. Upon accepting a sharing invitation, a viewers list and a message window appears on the right side of the *DeviceAnywhere Studio* interface.

Figure 3-13 shows the Viewers list on the right of the workspace.

Figure 3-13 Message Window

How to Send a Message:

1. Double click a name in the Viewers List.
2. Enter text in the Message window.
3. Click Send.

3.11 Exporting Frames and Video Media

DeviceAnywhere supports a real-time frame and video storage feature that captures the frames on a device during a Live session. From the time a device is acquired, frames are continuously uploaded to the server for every fifteen minutes of activity. Past fifteen minutes, new frames are uploaded and overwrite the previous frames.

When the Export command is selected, DeviceAnywhere enters a frame viewing mode. In this mode, you have the option to save one or more frames to your desktop. You can also upload a video clip to your MyDeviceAnywhere account.

Figure 3-14 shows the Frame Viewer. In this mode, you select the frames to upload by highlighting them with the cursor.

Notice the timeline below the frames. When you pass the Time Marker over a specific frame, all the frames generated during the time interval are displayed as a video clip.

Figure 3-14 shows the Frame Viewer.

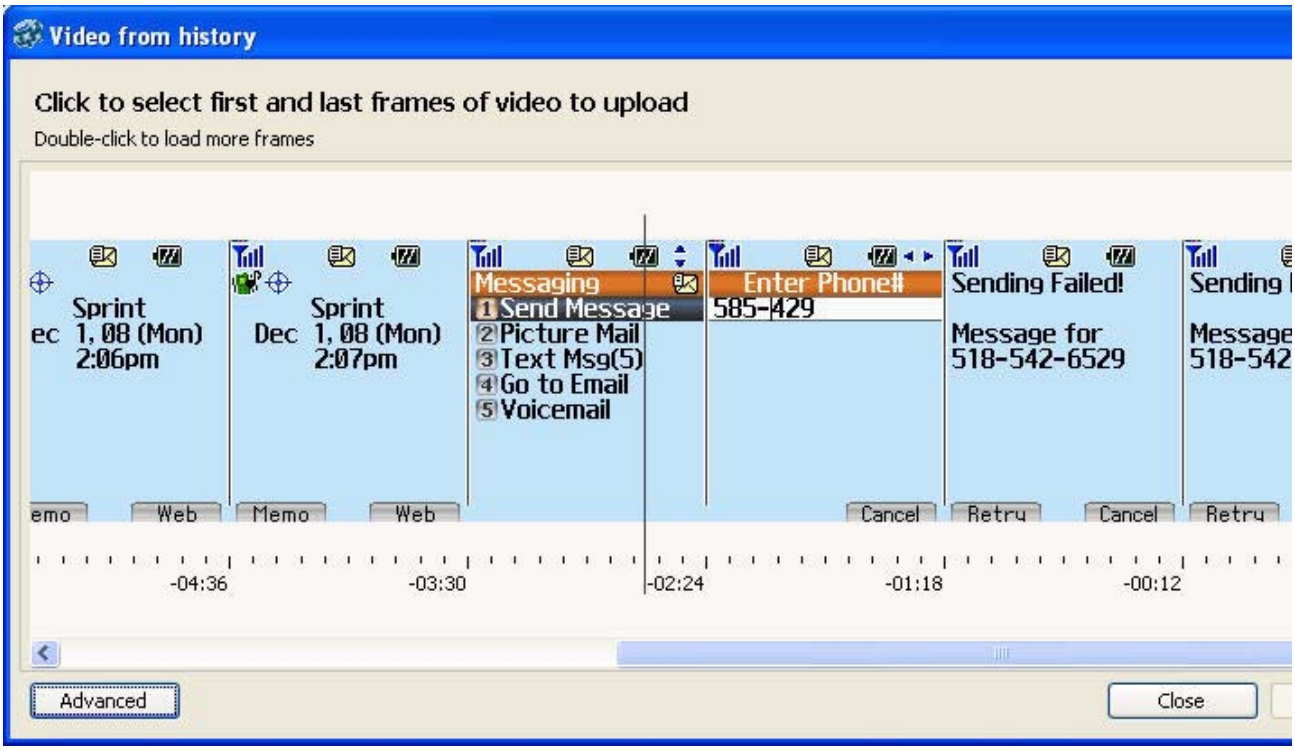


Figure 3-14 Frame Viewer

3.11.1 Video History

In the Frame Viewer, frames are organized into time segments in a two-tier hierarchy.

- Top Level—Frames at the top level are organized at about 1 minute intervals. For example, the second frame in Figure 3-14 has a start time of -04:36 and an end time of -03:30. This is a high-level view of device activity.
- Bottom Level—The frames within a time interval can be viewed by moving the Time Marker over a specific frame. All the frames taken during the time interval are played like a video clip. When you double click on a frame, the frames within this time interval are promoted to the top level.

3.11.2 Export Commands

The Export command in the Device menu gives you four exporting options:

- Current Frame—This option enables you to export one frame at a time to your desktop. You can export historical frames by using the Device Video controls to rewind to the desired frame.
- Frame from History—This option enables you to export many frames to your desktop. It's a good idea to review the recorded frames and select only the frames that you need for testing.
- Video from History—This option enables you to generate a video from the frames automatically captured.
- Upload Frames—This option enables you to upload frames to your MyDeviceAnywhere account.

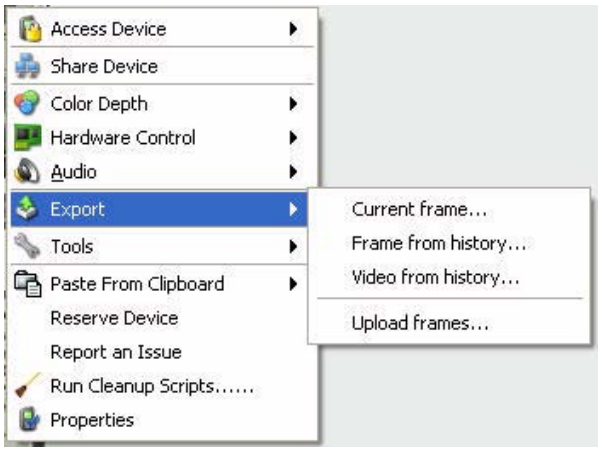


Figure 3-15 Export Command

The procedures below describe how to export frames.

3.11.3 Export the Current Frame

How to Export the Current Frame:

1. Navigate to the frame you want to export by using the Device Video Controls below the device.
1. Right click the device in the workspace. The Device menu appears.
2. Select Export.
3. Select Current Frame. The Save File dialog box is displayed.
4. Select the folder where to save the frame on your computer.
5. Enter the name of the frame.
6. Click the Save button.

3.11.4 Upload Frames

Use the Upload Frames option to upload a set of frames (in the DVR memory) to your *MyDeviceAnywhere* account. Figure 3-16 illustrates the Video from History dialog screen.

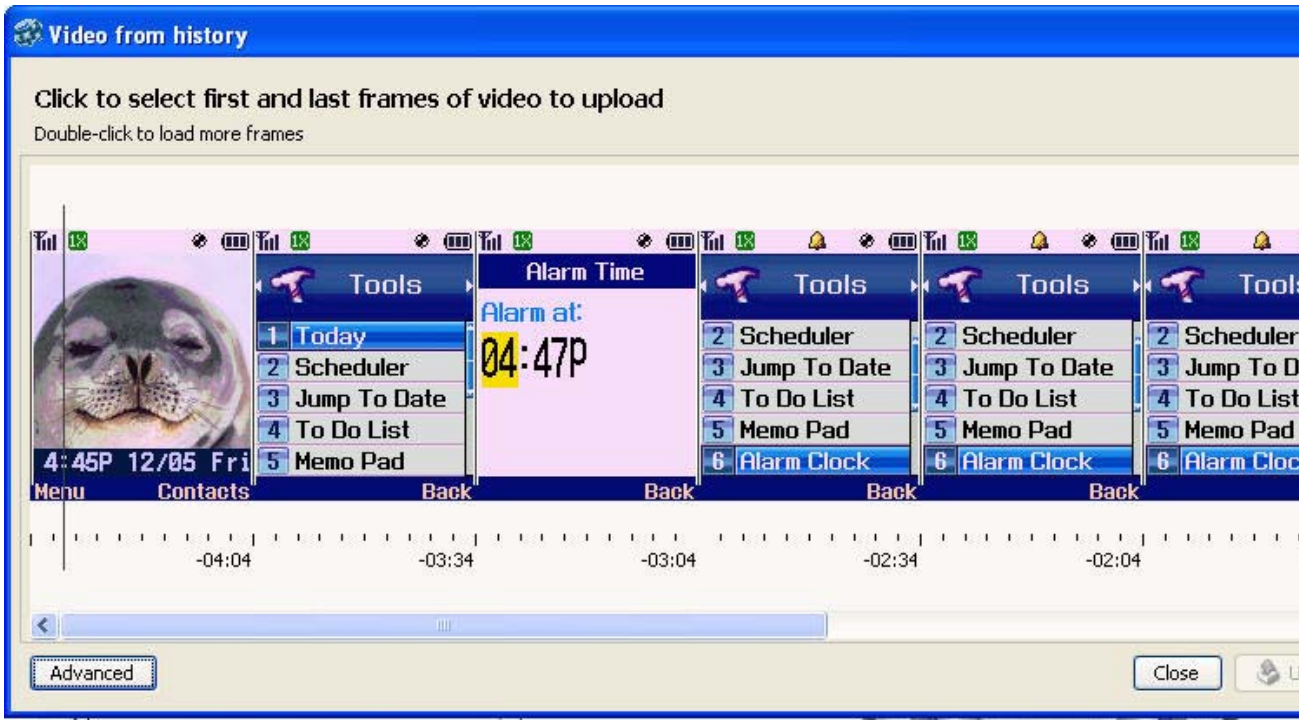


Figure 3-16 Video from History Dialog

How to Select and Upload Frames:

1. After entering key presses on the device, right click the device. The Device menu is displayed.
2. Select Export.
3. Select Upload frames. The device activity is displayed in the Frame Viewer.
4. Review the top level frames.
5. Double click a frame to mark the start frames.
6. Review the bottom level frames by passing the Time Marker over each top level frame.
7. When you want to promote bottom level frames to the top level, double click the frame. A loading bar is displayed.
8. Double click the frames to mark the stop frames.
9. Click Upload.
10. Enter the Recording name for this frameset.
11. Click OK.

After the frames are uploaded, a prompt appears asking if you would like to view the frames. When you select yes, you will be directed to MyDeviceAnywhere, where all uploaded frame results are stored under the Uploaded Frames tab.

3.12 Viewing Uploaded Frames in *MyDeviceAnywhere*

When you log in to *MyDeviceAnywhere*, you gain access to uploaded frames through the Uploaded Frames tab. This tab also provides the functionality to share results and collaborate with other users.

How to View Uploaded Frames in *MyDeviceAnywhere*:

In a browser, navigate to your *MyDeviceAnywhere* account.

1. Click the Upload Frames tab.
2. Click on the Name to view the uploaded frame details.

DeviceAnywhere Studio provides a report of all images in the frame set. Each image is numbered consecutively. The Data column shows one or more key presses associated to the image along with the Frame Annotations entered for individual frames. Figure 3-17 illustrates a detailed report.

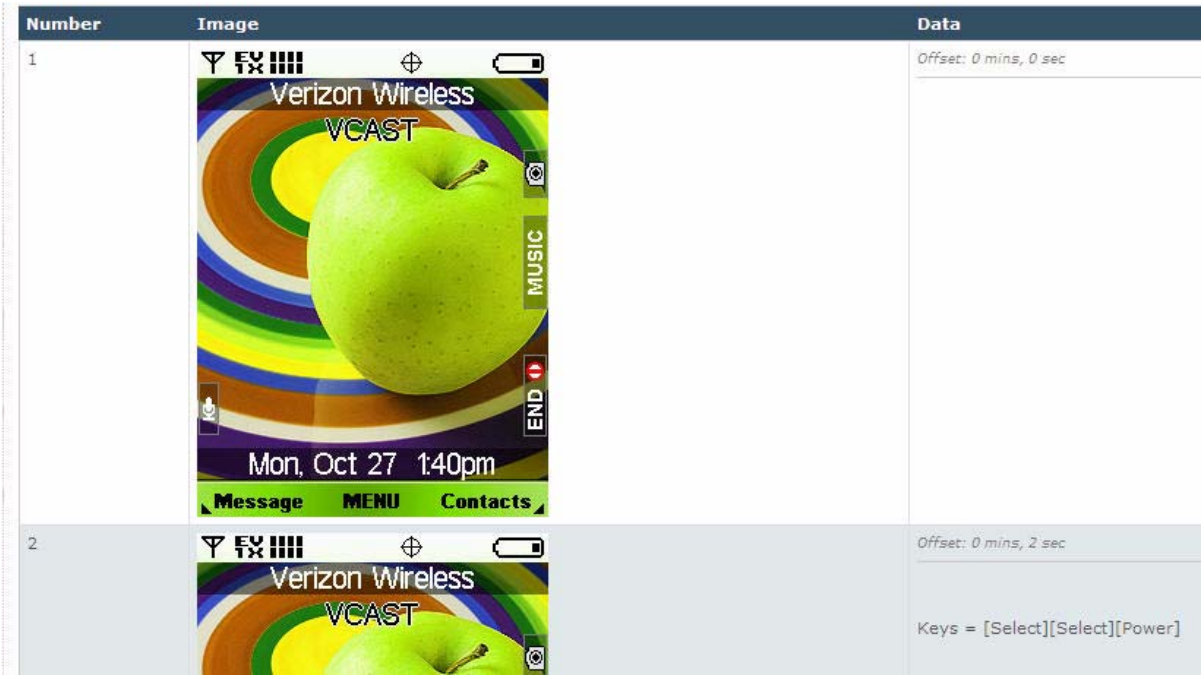


Figure 3-17 Upload Frame Details

In Upload Frame Details, two Email icons allow you to share results with other users. One icon (with lock) is used to send a secured email to a user in your project. The other icon is used to send an unsecured email to any user. After clicking the Email icon a form letter is generated with a link that enables the recipient to view the Uploaded Frame Results.

How to Share Results with other Users:

1. In the Upload Frame Details screen, click on an Email icon in the top right corner. The Email screen is displayed.
2. Enter the email address where you are sending results.
3. Select the names of any other colleagues who need to view results.
4. Click Send (top right).

Figure 3-18 illustrates the form letter that is sent to users. The letter provides a link to the *MyDeviceAnywhere* website where the results of uploaded frames can be viewed.

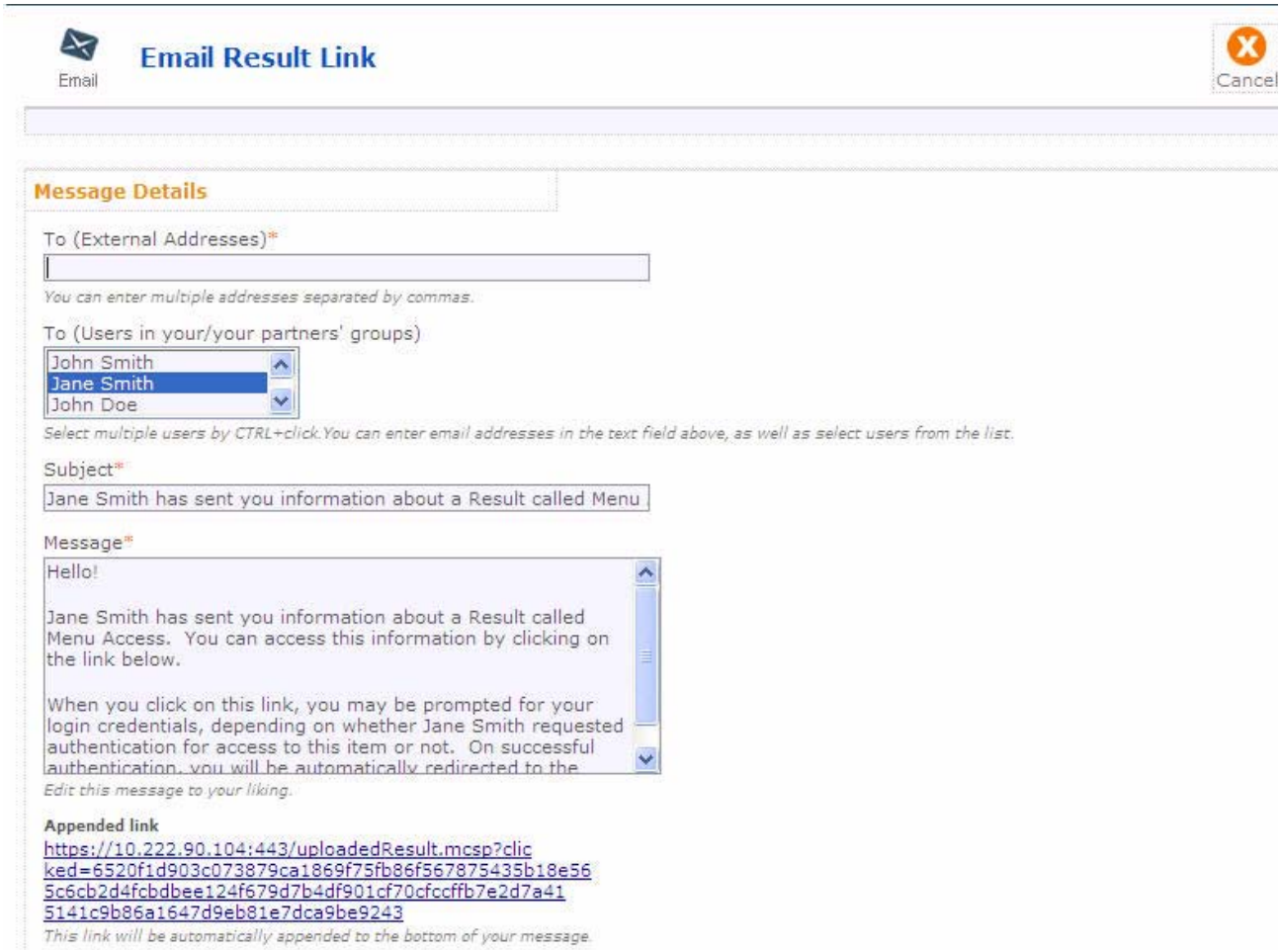

Email
Email Result Link
Cancel
Message Details
To (External Addresses)*
You can enter multiple addresses separated by commas.
To (Users in your/your partners' groups)
John Smith
Jane Smith
John Doe
Select multiple users by CTRL+click.You can enter email addresses in the text field above, as well as select users from the list.
Subject*
Jane Smith has sent you information about a Result called Menu
Message*
Hello!
Jane Smith has sent you information about a Result called Menu Access. You can access this information by clicking on the link below.
When you click on this link, you may be prompted for your login credentials, depending on whether Jane Smith requested authentication for access to this item or not. On successful authentication, you will be automatically redirected to the
Edit this message to your liking.
Appended link
https://10.222.90.104:443/uploadedResult.mcsp?clicked=6520f1d903c073879ca1869f75fb86f567875435b18e565c6cb2d4fcbdbee124f679d7b4df901cf70cfccffb7e2d7a415141c9b86a1647d9eb81e7dca9be9243
This link will be automatically appended to the bottom of your message.

Figure 3-18 Email Form Letter

For further information on your *MyDeviceAnywhere* account, refer to "MyDeviceAnywhere" on page 279.

3.13 Managing Wireless Applications

Most applications can be uploaded to the device over-the-air by doing the following:

- Sending a link to the device
- Entering the URL where the application is hosted

3.13.1 BREW Devices

DeviceAnywhere Studio enables you to upload BREW applications to a device by connecting to the BREW desktop. Look in the Properties screen of a device to determine whether it supports a BREW platform. When BREW is supported, the BREW menu is available through the Device menu.

BREW applications must be uploaded over a data cable. *DeviceAnywhere Studio* offers this capability through the BREW menu.

How to Access the BREW Desktop:

Right click on the device and select BREW in the Device menu.

Figure 3-19 illustrates the BREW menu.

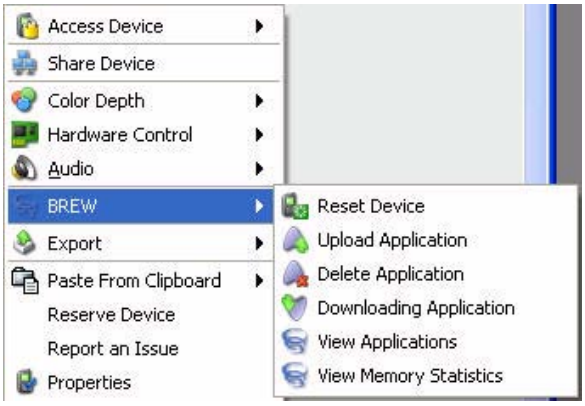


Figure 3-19 BREW Menu

The BREW menu provides the following options:

- Reset Device—Once you have uploaded your application, Reset Device power cycles the device so that you can then view your uploaded application.
- Upload Application—The Upload Application option enables you to upload the applications.
- Delete an Application—The Delete Application option enables you to delete the applications from the device.
- Downloading Application—The Downloading Application option enables you to download an application from the device.
- View Applications—The View Applications option enables you to view available applications.
- View Memory Statistics—The View Memory Statistics option enables you to view the memory status of the device.

The following procedures describe how to upload or download BREW applications.

How to Upload a BREW Application:

1. Right click on the device. The Device menu is displayed.
2. Select Hardware Control.
3. Select Connect Data Cable. This action connects the data cable of the device.
4. Right click on the device. The Device menu is displayed.
5. Select BREW.
6. Select Upload Application. The Upload BREW application dialog box appears.
7. Enter the name of the application to be uploaded (case-sensitive).
8. Select the local directory for all the application files when prompted.
9. When the Uploading BREW application screen shows the BREW file has uploaded successfully, click close to dismiss the screen.

Note: Ensure that the name of your application corresponds to your .mif file.

Use the View Applications command in the BREW menu to verify that the BREW file was uploaded.

How to Download a BREW Application:

1. Right click on the device. The Device menu is displayed.
2. Select BREW.
3. Select Downloading Application. The Download BREW file dialog box appears.
4. Enter the name of the file you want to download (case sensitive, and corresponding to the .mif file).
5. Enter the name and path of the file in the File screen.

Note: You can only download applications that you have uploaded yourself.

4 Record & Playback

4.1 Record & Playback Overview

DeviceAnywhere Studio includes a simple script-building function called Record & Playback. This function is available by clicking the Record & Playback icon in the left Icon Bar. This easy-to-use scripting feature allows you to record a series of steps on a device into a script. This functionality is for users who intend to create only simple scripts, and thus do not need some of the more complex scripting logic such as looping, branching, or macros.

Record & Playback is based on automatic capture. You press the Record button, press keys on a live device, and insert a WaitImage command when you want to insert a checkpoint into the script. Thus, while recording you alternate between pressing keys on the device and clicking the WaitImage icon on the toolbar. When you are done, you click the Record button again to stop recording.

Record & Playback supports the commands you need to organize scripts, edit them, change properties, save changes, and upload test results to your *MyDeviceAnywhere* account.

Figure 4-1 shows the Record & Playback view.

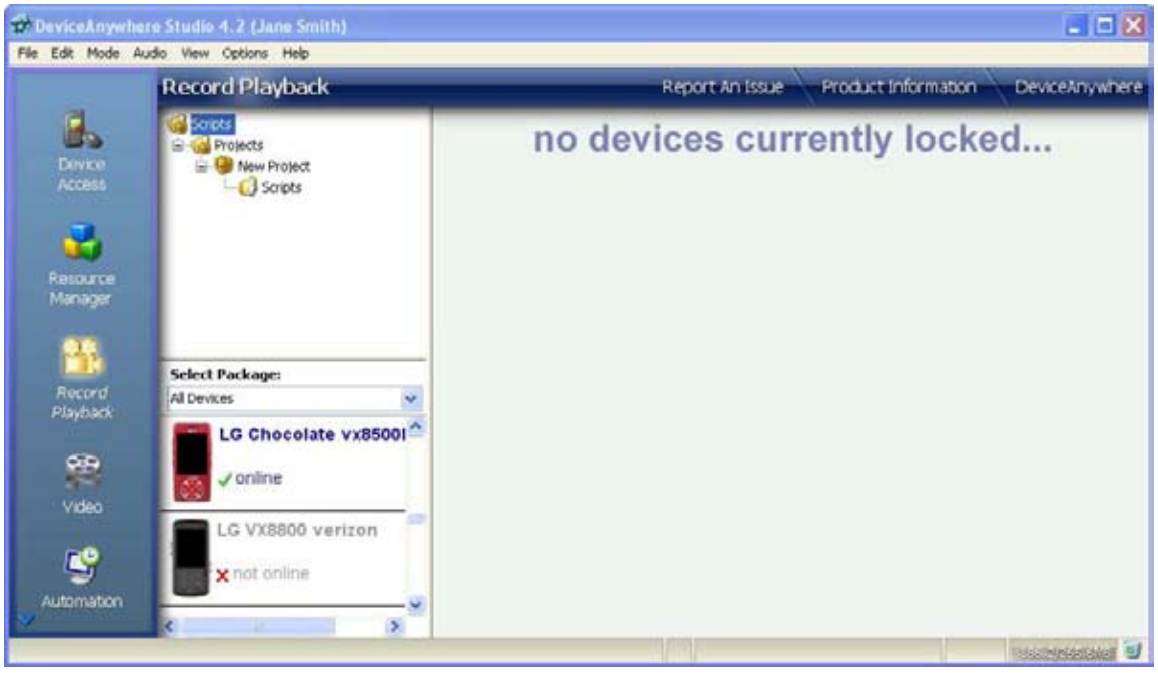


Figure 4-1 Record and Playback View

4.2 Get Ready to Record a Script

This section describes how to start a new script and record the key presses on a device in order to create a new script.

4.2.1 Manage Folders and Scripts

The Explorer provides project folders where you can organize your scripts.

All scripts are part of a Project. A default project is automatically created for all accounts when *DeviceAnywhere Studio* is launched. Notice the DA-Project in the Explorer. To organize your work for a project, it's often useful to create a folder and then create scripts within the folder for different tests.

How to create a new folder and new script:

1. Right click on the Scripts folder under DA-Project in the Explorer.
2. Select New Folder.
3. Enter the name of the folder. In this case, enter Alarm.
4. Right click on the new folder.
5. Select New Script.
6. Enter the name of the new script. (This is often the name of the device under test.)

When a script is created, an empty script is displayed in the workspace. The red target indicates an empty script. Figure 4-2 illustrates an empty script before a device is added to the script.

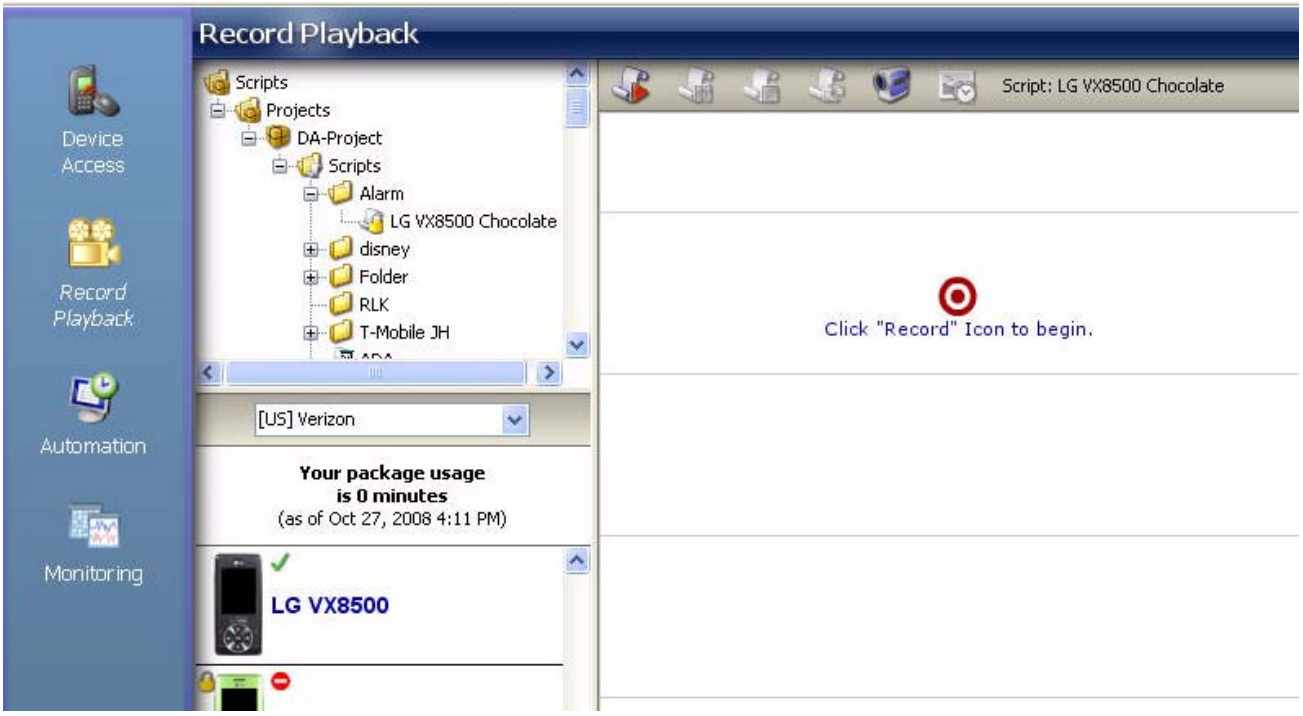


Figure 4-2 Empty Script

4.2.2 Add a Device to the Script

You must add a device to the script before recording a script. In Record & Playback, only one device can be added to a script at a time. When you add a device, the device is displayed in the far right panel of the screen.

The procedure below adds a device to the script.

How to Add a Device to a Script:

1. Right click on a device in the Device List to add the device that you want to use.
2. Select Add to Script.

When you want to use another device in a script, you can remove the current device in the same way. Right click on the device, and the Device menu is displayed. This time select Remove from script.

Figure 4-3 shows the Device menu with Add to Script selected.

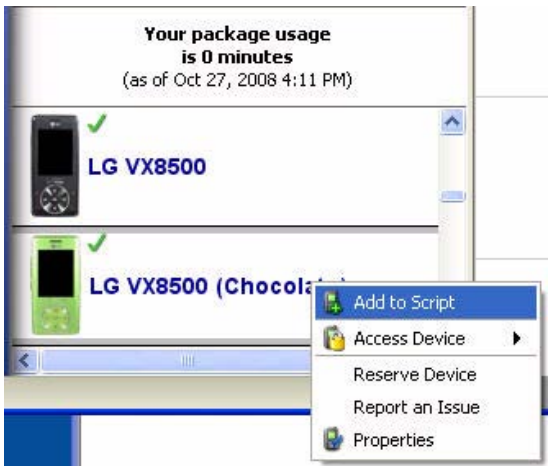


Figure 4-3 Add to Script

4.2.3 Access and Acquire a Device

You must access a device and acquire it before recording a script.

The procedure below describes how to access a device and acquire it.

How to access and acquire a device:

1. Right click on the device in the far right panel. The Device menu is displayed.
2. Select Access Device.
3. Select Acquire Device.

For further information on accessing and acquiring a device, refer to "Device Access" on page 27.

Figure 4-4 shows the Device menu with Acquire Device selected.

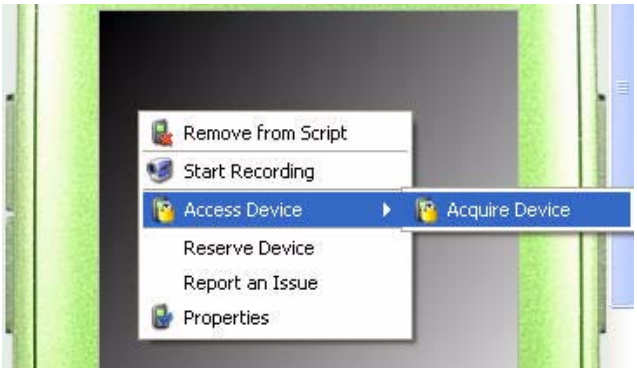


Figure 4-4 Acquire Device Menu

Once all the steps in this section are completed, you are ready to record a script.

Figure 4-5 illustrates the Record & Playback view when a script is ready to record.

Notice the name of the script in the toolbar. In this case, Script LG VX8500 will be recorded. A device is shown in the far right panel. The LG VX8500 was accessed and acquired to record the LG VX8500 script. The device screen is showing the carrier idle screen. This screen indicates the device is active and ready to receive input.

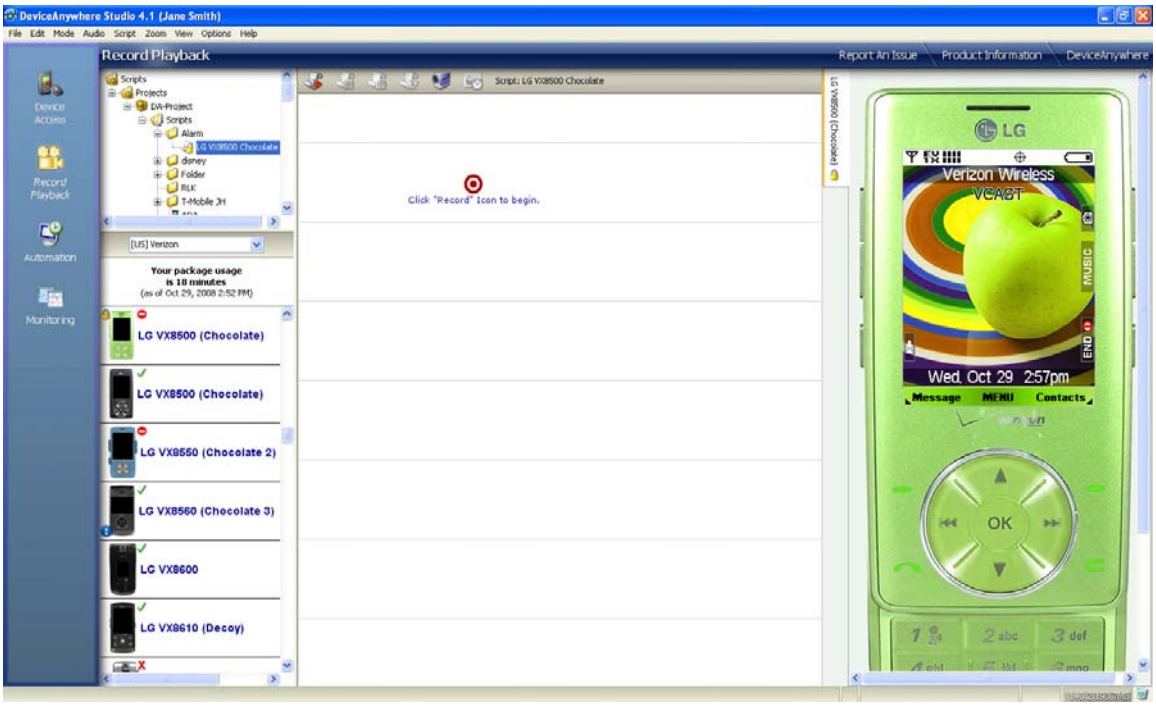


Figure 4-5 Record & Playback Script—Ready to Record

Note: Before you start recording make sure that the device is awake and accepting input. Most devices go into a power save mode after a certain amount of inactivity. This amount of time varies per device.

4.3 Script Commands

In this section, you receive a brief introduction to the SendKeys and WaitImage script commands. These commands are the most commonly used commands in *DeviceAnywhere*.

4.3.1 SendKeys Command

The SendKeys command can be used to execute a set of key presses on a device. The main edit field in the Properties screen, Test To Send, is used to enter the keys to send. This edit field can contain numbers, letters, URLs, and control keys such as [SoftR] or [SoftL].

Variables can be used to define the keys being sent to the device. To insert a variable into a string, click on the Puzzle piece next to the key edit field.

The correct key mode should be selected for the keys being sent to the device. For control keys, the mode generally doesn't matter. However, for numbers and letters, it is important to know the current device input mode. For example, on a SMS send screen, a device is set to Alpha mode, expecting text characters to be entered into the message. When defining the phone number to

send the message to, the device is set to Numeric mode. Some devices map special characters, such as colon (:), question mark (?), or forward slash (/) differently when a web browser is running. In this case, the Web mode (URL) is used.

In Record & Playback, the SendKeys command captures reference images (both before and after the key press). Once a SendKeys is inserted and associated to a reference image in the script, the SendKeys can be reconfigured to reference a different image.

SendKeys commands are automatically inserted into a script as you press keys on a device that's acquired.

When a SendKeys is inserted into the script through key presses, *DeviceAnywhere* associates one key press with one SendKeys command. Multiple key presses cannot be recorded in one SendKeys through automatic capture. However, you can manually enter key presses into the SendKeys Key field. Refer to Edit the SendKeys Key field below.

For further detail on the SendKeys command, refer to "SendKeys Properties" on page 70.

4.3.2 WaitImage Command

The WaitImage command waits for a specified image to appear or times out when the image is not found. A timeout can end the script immediately, based on the Timeout Action selected in the WaitImage Properties screen.

The Timeout setting in the WaitImage Properties screen is important to successful script execution. The Timeout setting specifies the number of seconds to wait for the image to appear. This setting must anticipate network response time. Otherwise, the WaitImage can return a failure message.

A WaitImage command is a checkpoint. What is a Checkpoint?

- Checkpoints are used to verify that your mobile application is functioning properly and serving the correct content. Checkpoints stop the execution of the test script temporarily. Without checkpoints in your script, you could send a series of key presses to the device in the SendKeys command, but you could never check to see whether you are getting the expected responses. Checking for expected responses is an important part of testing.
- Checkpoints are also important to help synchronize the test scripts. If you are testing network applications and you are uncertain of a transaction response time, you must include checkpoints that force the script to wait for the device to respond before moving to the next step.

During a session when you are recording key presses on the device (SendKeys are appearing in the script), you can insert WaitImage commands by clicking the WaitImage icon in the toolbar. *DeviceAnywhere* must be in the Record mode to insert a WaitImage command.

When you click the WaitImage icon, you are automatically capturing a reference image (as an entire screen) in a WaitImage command in the script. The WaitImage command can wait for the entire screen to match or can search for a region of the screen. Typically, it is best to choose a region of the screen.

For instance, since text that displays the time and network strength meters constantly changes, no match would ever succeed. Therefore, time and network strength meters should be excluded

from an image compare function. In addition, matching an entire screen takes more processing time than a match to a region of the screen.

You should also consider color reduction control to facilitate matching of images across devices with different color depths. Refer to Color Depth in the Device menu in “Device Menu” on page 29.

You’ll have to work fast at times to click the WaitImage icon, because the WaitImage captures the current frame on the LCD, which might not always correspond to the image you want as the reference image for the WaitImage command.

You can’t go back and edit the reference image in a WaitImage command. Delete a WaitImage that you don’t want in the script. Just select the WaitImage with your cursor and press the Delete button on your keyboard. You may have to go back and re-record your steps.

For further detail on the WaitImage command, refer to “WaitImage Properties” on page 74.

4.4 Create an End-to-End Script

Make sure you have created a script, added a device, and acquired the device before you start recording. This section describes how to create an end-to-end script by using the recording controls in the toolbar above the script editor.

Figure 4-6 shows an empty script in the script editor. Record and playback controls are provided as icons on the toolbar.

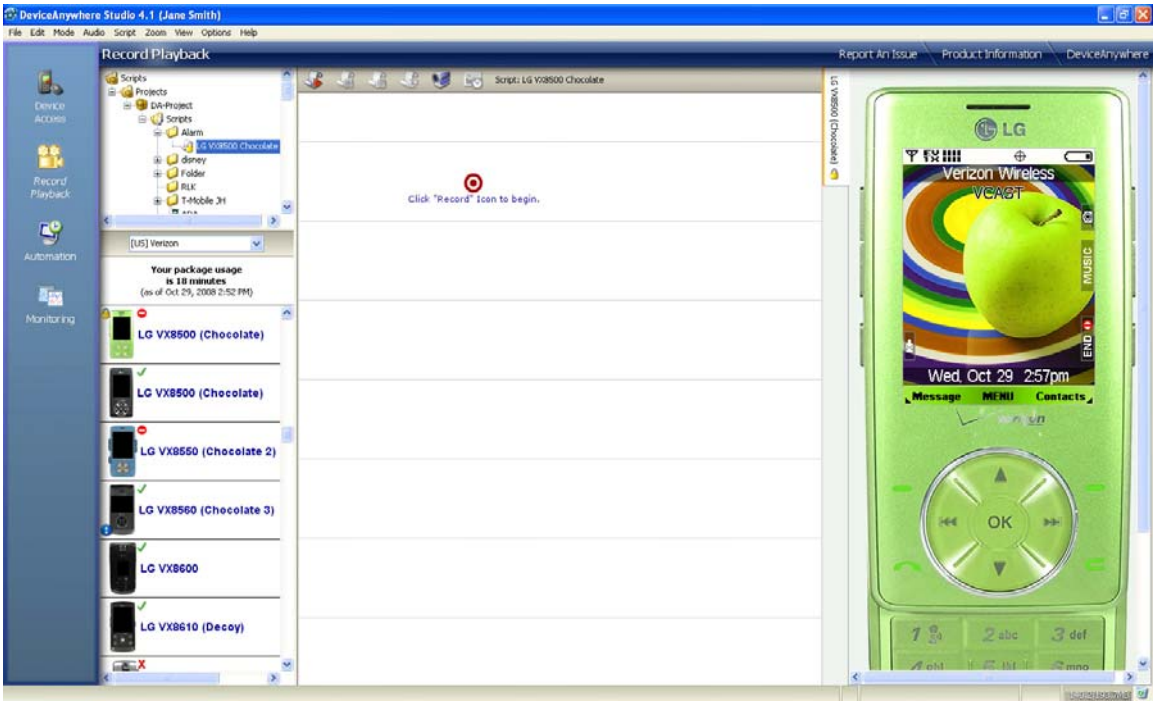


Figure 4-6 Empty Script

4.4.1 Record and Playback Controls

The Record and Playback toolbar has five controls (from left to right):

- Play Script—This control executes a script. Play Script is disabled when a script is already executing.
- Pause Script—This control enables pause of an execution of a script. Pause Script is only enabled during script execution.
- Stop Script—This control stops the execution of a script. Stop Script is only enabled during script execution.
- Restart Script—This control allows you to restart the execution from the start. Re-Run Script is available during script execution.
- Record Script— Click on Record Script, and a recording light blinks on the button indicating that recording has begun.

You can record a script by pressing the Record button and pressing keys on the device. When necessary, click the WaitImage icon in the toolbar to insert a WaitImage command. All your interactions with the device are recorded as a script.

To stop the recording, simply press the Record button again. The blinking light stops, indicating the recording has stopped.

4.4.2 Sample End-to-End Script—Set Alarm

The steps needed to record a simple script—Set Alarm are provided in Table 4-1. These steps are somewhat generic, since each device in a network has its own unique set of key presses to set the alarm.

Table 4-1 Set Alarm Script

Action	Step	Expected Result
1. Start recording	Click the Record button in the toolbar.	Start the recording.
2. Return to idle screen	Press the Power/End key on the device.	Device returns to idle screen. The device is reset.
3. Open main menu	Press the OK key on the device.	Device goes to the main menu.
4. Press WaitImage icon	Insert a WaitImage command in the script to capture a reference image of the main menu.	A reference image is captured of the main menu.
5. Navigate to Settings & Tools.	Press the left arrow key one time to navigate to Settings & Tools on the main menu.	The Settings & Tools is selected.
6. Open Settings & Tools.	Press the OK key on the device.	The Settings & Tools menu appears.
7. Press WaitImage icon	Insert a WaitImage command in the script to capture a reference image of the Settings & Tools menu.	A reference image is captured of the Settings & Tools menu.
8. Navigate to the Tools menu.	Press the down key to select the Tools menu.	The Tools menu is selected.

Table 4-1 Set Alarm Script (Continued)

9. Open the Tools menu.	Press the OK key on the device.	The Tools menu is displayed.
10. Press WaitImage icon	Insert a WaitImage command into the script to capture a reference image of the Tools menu.	A reference image is captured of the Tools menu.
11. Navigate to Alarm clock.	Press Down, Down on the arrow keys to navigate to the Alarm clock option.	The Alarm clock option is selected.
12. Open the Alarm clock	Press the OK key on the device.	The Alarm clock settings are displayed.
13. Select Alarm 1.	Press the OK key to select Alarm 1, which is the first option in the menu.	Alarm 1 settings are displayed.
14. Open the time settings.	Press the Left key to set the time.	The time setting entry field is displayed.
15. Change the hour.	Press 4 on the device to enter 4 for the hour.	The Alarm hour is changed to 4.
16. Go to the minute field.	Press the Right key on the device to navigate to the minutes field.	Enter 30 into the minute field.
17. Go to the AM/PM field.	Press the Right key on the device to navigate to the AM/PM field.	Enter PM.
18. Press WaitImage icon	Insert a WaitImage command in the script to capture a reference image of the Alarm settings screen.	A reference image is captured of the Alarm settings screen.
19. Confirm Alarm 1 settings.	Press the OK key twice on the device.	The device returns to the Alarms screen.
19. Return to idle screen.	Press the End key on the device.	Device returns to idle screen. The device is reset.
20. Stop recording	Click the Record button in the toolbar.	Stop the recording.

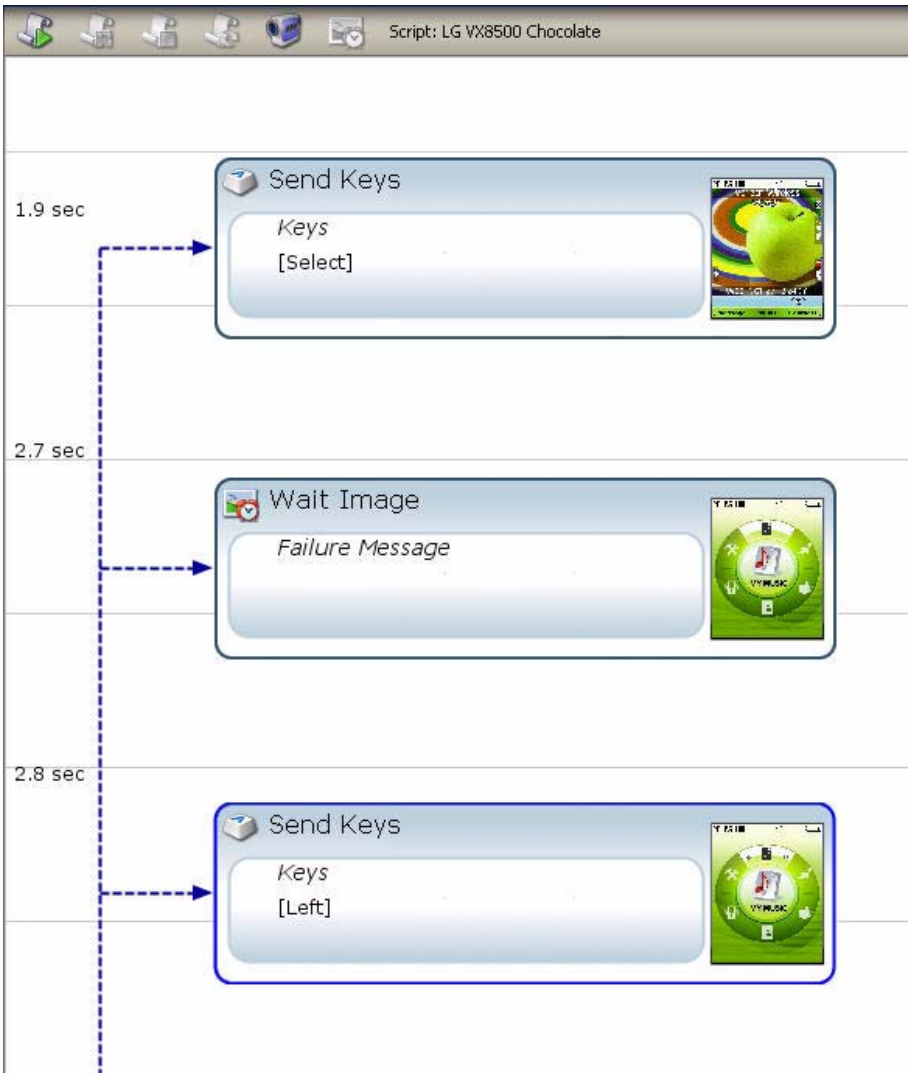


Figure 4-7 Set Alarm Script

4.4.3 What Your Automated Script Looks Like

The steps described in Table 4-1 were recorded in the Set Alarm script shown in Figure 4-7.

Each rectangle in the script shows a separate SendKeys or WaitImage command.

Key presses are recorded into separate SendKeys commands. For instance, when you press [End], one SendKeys is added to the script. However, you can manually enter more key presses into the key field to capture multiple key presses in one SendKeys command. Refer to "SendKeys Properties" on page 70.

Wait Time

The Wait Time corresponds to the amount of time you paused during script recording. If you pause between actions while recording, you see more Wait Time on the left edge of the script.

How to Change the Wait Time:

If the Wait Time was unintentional or if it is not the length of time desired, click on the Wait Time and change it directly by entering a new value. You can select a rectangle with the cursor and move it up or down in the script editor. This action adjusts the Wait Time on the left edge of the script.

Edit a SendKeys Key Field

Click the space under Keys as a quick way to edit the key field without having to open the Send-Keys Properties screen. You can do this to enter more than one key press into a SendKeys. This key field is labelled Text to Send in the SendKeys Properties screen.

Edit a WaitImage Message Field

Similarly, click the space under Failed message as a quick way to edit the error message generated for the WaitImage without opening the Properties screen. The message field is labelled Detailed Message in the WaitImage Properties screen.

Edit the SendKeys Reference Image

The image showing on the right of the SendKeys is a reference image, a screen shot of the image on the device before and after a key press. This reference image can be edited through the WaitImage Tab in the SendKeys' Properties screen. Refer to "WaitImage Tab in SendKeys Properties" on page 71 for more information.

Delete a Script Command

How to Delete Commands in a Script:

One or more commands can be deleted by selecting the commands with your cursor and press-ing the Delete key on your keyboard.

4.4.4 Working with Scripts

While working on a script, you may need to save changes, discard changes to start over, or export the script to your computer. The Script menu provides these options.

4.4.5 Script Status

Scripts are available for editing or saved in the database. A yellow lock on the script icon indicates that the script is open for editing. No other user has access to the script in the database. After you save a script, the yellow lock is removed. Refer to “How to Open a Script for Editing” below.

4.4.6 Open the Script Menu

How to Open the Script Menu:

To open the Script menu, right click on a script in the Explorer.

The Script menu has the following options.

- Rename Script—Rename Script lets you modify the name of a script.
- Save Changes—Save Changes saves the script to the database.
- Cut Script—
- Paste Script—
- Export Script—The Export Scrip option allows you to export the script to your computer.
- Discard Changes—Discard Changes lets you get rid of all changes in the current script.
- Delete Script—Delete Script removes the script from the Explorer.
- Properties—This option opens the Script Properties screen. This screen provides a number of settings. It also allows you to export and import data sets to be used in scripts. Refer to “Script Properties” on page 66.

4.4.7 Perform Simple Tasks with the Script Menu

The following procedures describe how to use the Script menu to perform simple tasks.

Rename a Script

1. Right click on the script. The Script menu appears.
2. Select Rename Script. The file name is open for editing.

Save a Script

How to Save a Script to the Database:

1. Right click in the script editor. The Script menu appears.
2. Select Save Changes. The yellow lock is removed from the script icon in the Explorer.

Open a Script for Editing

How to Open a Script for Editing:

1. Right click on the script in the Explorer.
2. Select Lock/Edit Script. The yellow lock appears on the script icon in the Explorer.

Export a Script

How to Export a Script:

1. Right click on the script in the Script Editor.
2. Select Export. A Save file entry screen is displayed.
3. Navigate to the folder where you want the script saved.
4. Enter the name of the script.
5. Click Save.

The script is saved to your computer.

4.4.8 Script Properties

The Script Properties screen contains a number of settings to control and review the history of scripts in a test environment. These tabs allow you to add comments about a script. Four tabs are available in this view.

How to Open the Script Properties Screen:

1. Right click on a script in the Explorer. The Script menu appears.
2. Select Properties.

Figure 4-8 illustrates the Script Properties screen.

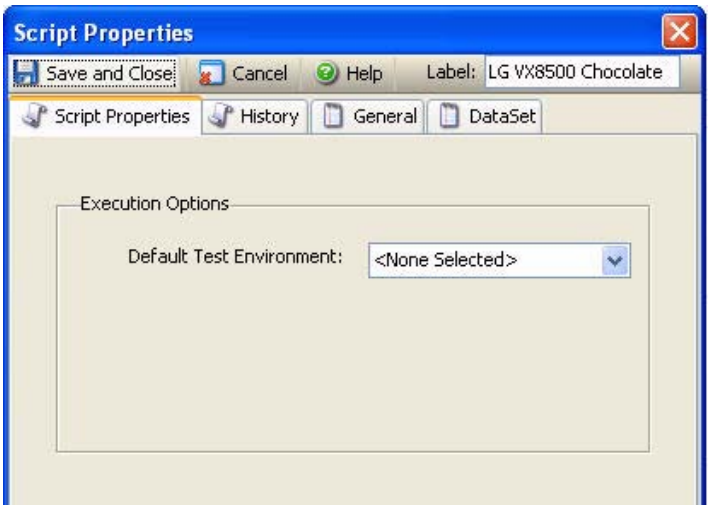


Figure 4-8 Script Properties Screen

4.4.9 Properties Screen Toolbar

The Properties screen toolbar at the top supports basic commands to save changes, cancel changes, and access online Help while working in the Properties screen. A brief description of these commands follows:

- Save and Close—This command lets you save the settings you entered into the Properties screen.
- Cancel—This command allows you to dismiss the Properties screen without saving any of the changes you made.
- Help—This command provides access to online Help.
- Label—The Label entry field lets you enter the name for each SendKeys.This is a useful feature to label the steps in a long script.

All Properties screens in *DeviceAnywhere Studio* contain this toolbar and function in the same way.

4.4.10 Default Test Environment

DeviceAnywhere Studio supports scripting efforts for three test environments:

- Production
- Staging
- Development

When you first create a script, you might want to set the test environment that the script is written for. The test environments represent the stages of development. Test environment is used like a global parameter. When the test environment is moved from one stage to another, the scripts can be promoted or validated for a new test environment.

4.4.11 Script History

The History tab provides access to all versions of the script stored in the database. From here you can checkout a previous version or delete a version of a script. Once the database is changed, these changes become permanent.

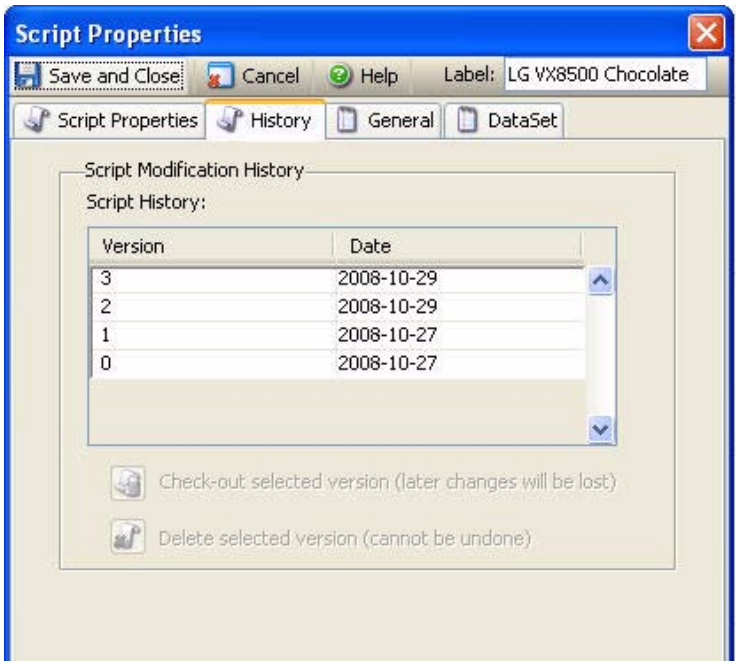


Figure 4-9 History Tab in Script Properties

How to Check Out a Previous Version of the Script:

1. Highlight a version of the script with your cursor.
2. Click the Check-out selected version button below the window.

The previous version will be displayed in the script editor.

4.4.12 Script File Properties

The General tab shows file properties such as creation date and the last date modified for a script. This tab also has an entry field to enter comments.

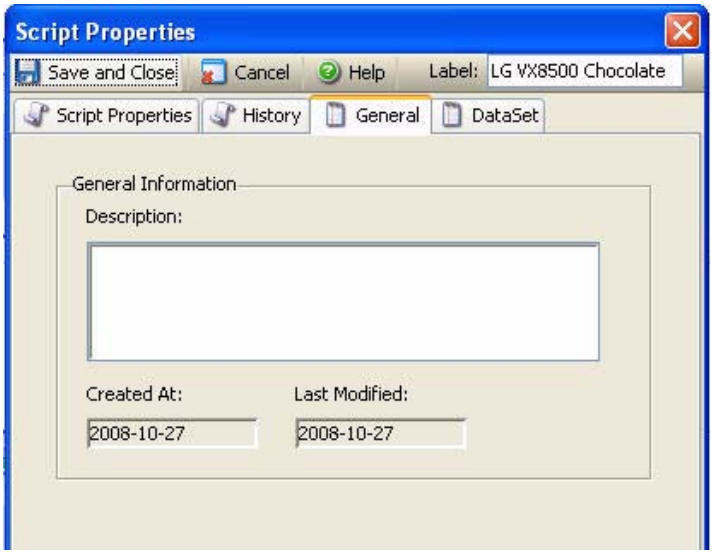


Figure 4-10 General Tab in Script Properties

4.4.13 Dataset

This tab is used to create datasets and import files in .csv format. Figure 4-11 shows a list of phone numbers and text messages. Phone Num and Message are defined as variables in the DataSet tab. When the variables are evaluated during script execution, the phone numbers and messages are used in the Send SMS script.

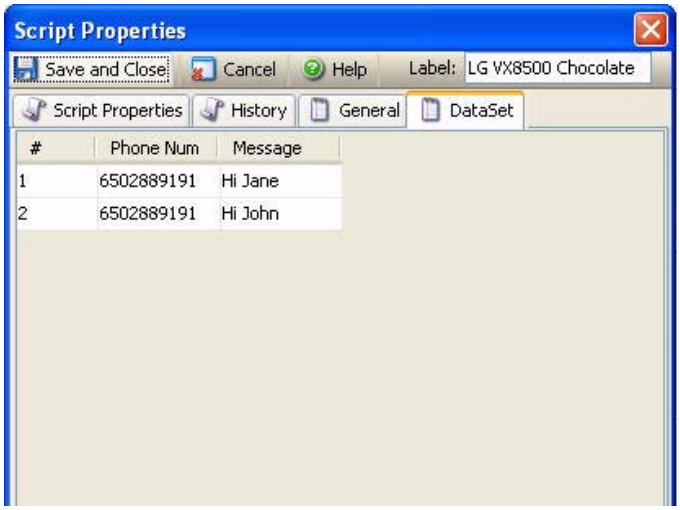


Figure 4-11 Dataset Tab in Script Properties

4.5 SendKeys Properties

Figure 4-12 illustrates the SendKeys Properties screen.

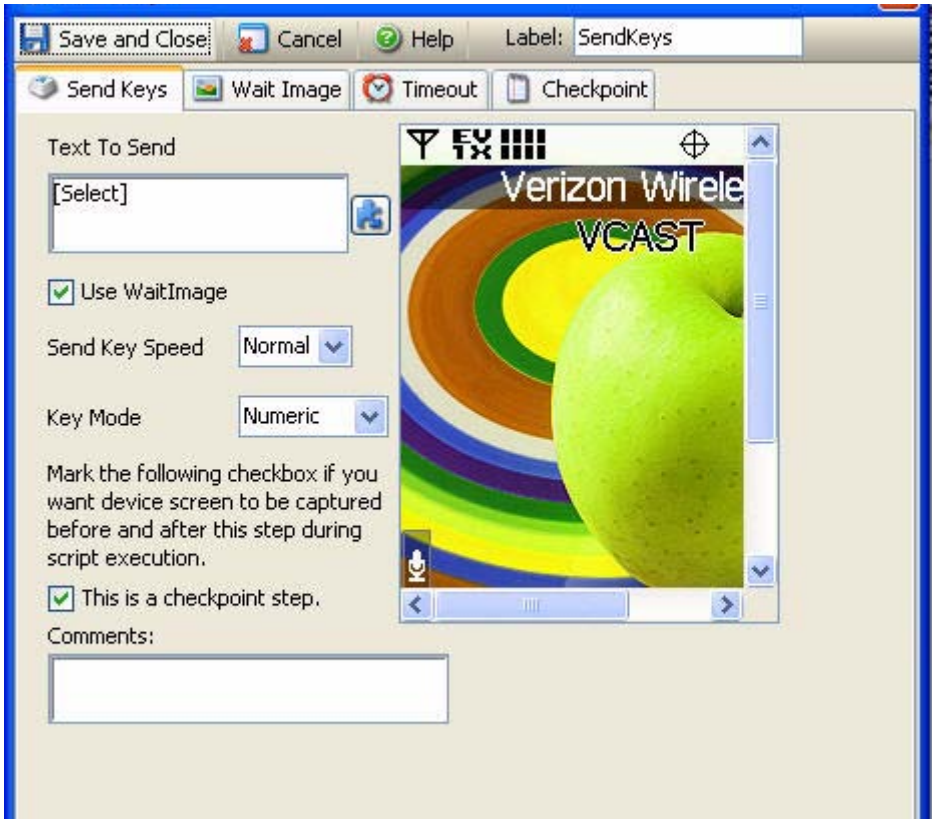


Figure 4-12 SendKeys Properties Screen

SendKeys configuration settings include entering data in the Text To Send field, Selecting the Key mode, and using a WaitImage or Checkpoint.

How to Open the SendKeys Properties Screen:

To open the SendKeys Properties screen, double click on the SendKeys. The Properties screen is displayed.

The SendKeys Properties screen has the same toolbar as the Script Properties screen described above. Refer to “Properties Screen Toolbar” on page 67 for more information.

4.5.1 SendKeys Properties

The Properties tab (default view) provides the following settings:

- Text To Send—The Text to Send field contains the key press associated to the SendKeys.
- Key Mode—This field lets you select the key mode: Alpha, Numeric, or Web URL, from the drop-down menu.
- User WaitImage Checkbox—When this box is checked, two new tabs, WaitImage and Timeout, are automatically available in the Properties screen.
- This is a checkpoint step Checkbox—When the checkpoint checkbox is selected, the images that appear on the device screen before and after this step in script execution are captured.
- Comments—Use this field to add comments about this SendKeys.

4.5.2 WaitImage Tab in SendKeys Properties

Figure 4-13 shows the WaitImage tab in the SendKeys Properties screen. This tab allows you to edit the reference image of the SendKeys command.

Figure 4-13 WaitImage Tab in SendKeys Properties

The WaitImage tab provides the following settings:

- Selection Controls—This button is used to select a region of the reference image for the compare function. Click Selection Controls, then draw a rectangle on the image on the left to indicate the region that will be used in the compare.
- Image Location—This button is used to specify criteria to compare the reference image with the device screen. The Fixed setting is based on an exact coordinate location. The Any Position setting provides for tolerance in matching the images and results more often in a successful match.

4.5.3 Timeout Tab in SendKeys Properties

Figure 4-14 illustrates the Timeout tab for the SendKeys command when Use WaitImage is checked.

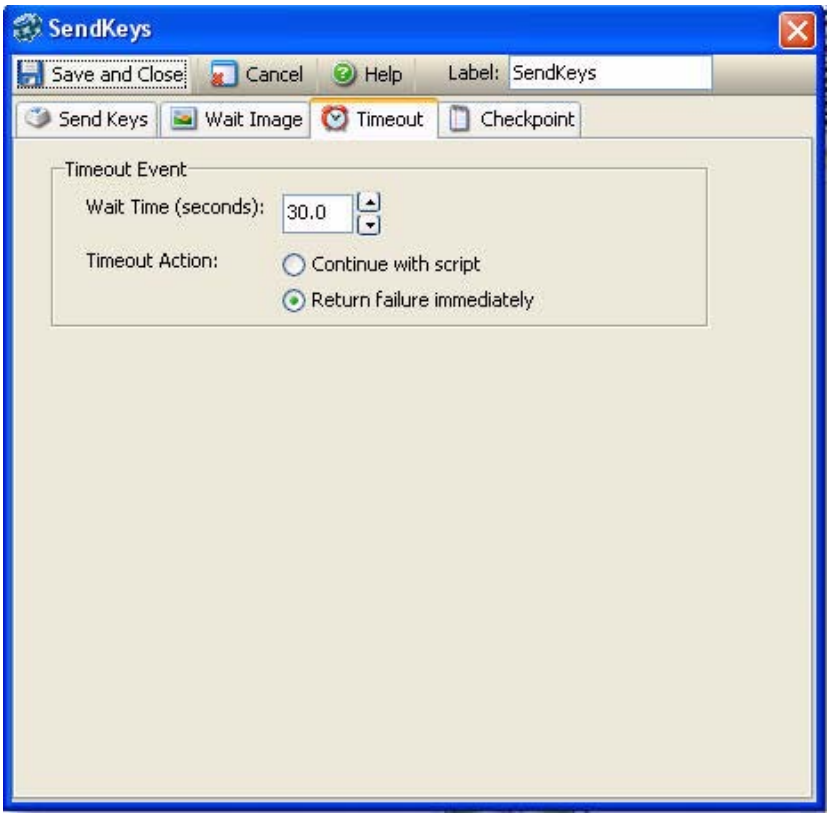


Figure 4-14 Timeout Tab

The Timeout tab provides the following settings:

- Wait Time—Wait Time specifies how many seconds the command should wait for a defined action. For example while waiting for an image to download, you can add a timeout of 8 seconds. If the image does not load in 8 seconds, *DeviceAnywhere Studio* will declare a timeout.
- Time Out Action—Timeout Action allows you to specify the action to take in the event of a timeout. You can either (a) continue with the script or (b) return failure immediately.
  - In case of a timeout, the Continue with the Script option does not break the script execution. It continues with the next command. For example, while waiting for an image to load, if the command experiences a timeout, you may choose to continue with the script and execute the remaining commands.
  - The Return Failure Immediately option allows you to stop script execution in case of a timeout. This may be required when the remaining script depends on the successful completion of this command. For example, if your script loads a WAP page, clicks on a link, and then performs tasks on the subsequent page, you can use the Return Failure Immediately option at the point where you click on the link. This terminates the script when the click on the link action times out.

How to Delete a SendKeys in the Script

1. Click on the SendKeys to be deleted.
2. Press the Delete key on your keyboard.

How to Enter Multiple Key Presses in a SendKeys:

1. Click the field directly under Keys on the SendKeys. The Text to Send field is opened.
2. Place your cursor in the field.
3. Type your entry on your keyboard. Remember to use the DeviceAnywhere key stroke syntax. Each command must be enclosed in brackets.

4.5.4 WaitImage Properties

Figure 4-15 illustrates the Properties screen for a WaitImage.

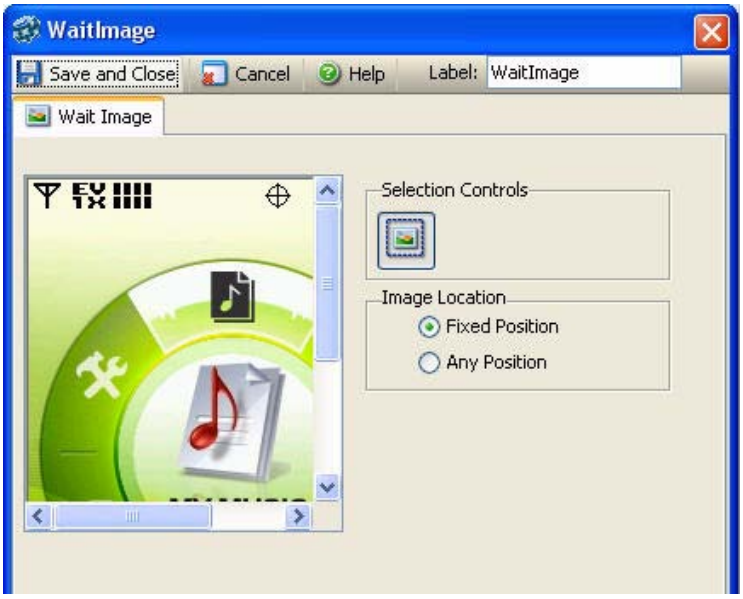


Figure 4-15 WaitImage Properties Screen

The WaitImage Properties screen has a number of settings to configure the WaitImage. It allows you to select an area of the reference image that will be used in the compare function. It also allows you to indicate where *DeviceAnywhere* scans for the compare function.

How to Open the WaitImage Properties Screen:

To open the WaitImage Properties screen, double click on the WaitImage. The WaitImage Properties screen appears.

The WaitImage Properties screen has the same toolbar as the Script Properties screen. Refer to "Properties Screen Toolbar" on page 67 for more information.

The following settings are available on the WaitImage Properties screen:

- Selection Controls—This control is used to set the reference area.
- Fixed—Click the Fixed position when the matching area in the device image must appear exactly in the same location in the reference image.
- Any Position—Select Any Position when the matching area in the device image is the same as the reference image, but it's position does not need to be exact.

4.5.5 Set the Reference Image

By default, when a WaitImage command is inserted into a script, the entire screen is used for the compare function. The procedure below is useful in setting a region of the reference image in a WaitImage command. It is recommended for all WaitImage commands in your script, since it shortens processing time for the compare function.

How to Set the Reference Image:

1. Double click the WaitImage. The WaitImage Properties screen appears.
2. Click Selection Controls.
3. Draw a rectangle over the image (on the left) with your cursor to indicate the region of the image that will be used in the compare function.This included region is highlighted.
4. Choose the location of the search: Fixed or Any Position (which scans the entire image for a match).
5. Click Save and Close.

The WaitImage is set to a region of the reference image for a compare function.

4.6 Play Script

When a script is open in the script editor, press the Play button in the toolbar. Each step is executed and highlighted on the screen with a blue rectangle.

When you are starting up, click on the script in the Explorer to open it in the script editor. Click the Play button on the tool bar to run the script.

4.7 Upload Test Results

Whenever you complete running a script, you see a Script Successful message or a Script Failure message.

When you are satisfied with the script, you can upload the test results to your *MyDeviceAnywhere* account. The Script Status screen provides an option to upload.

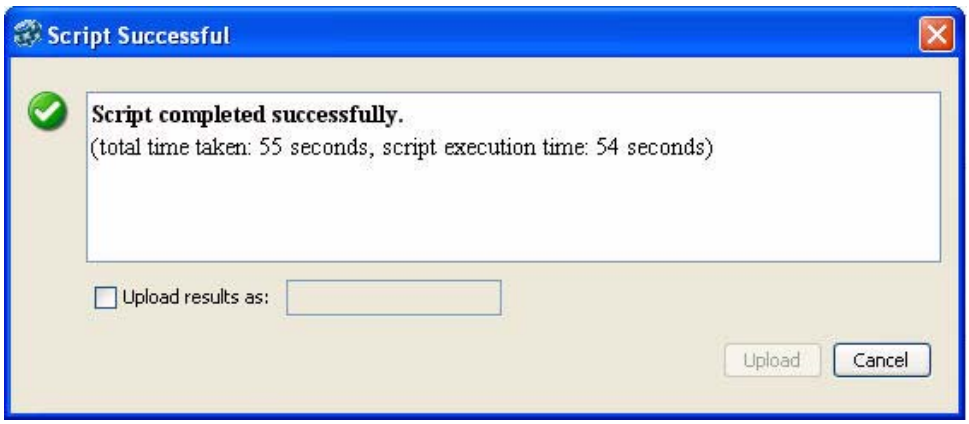


Figure 4-16 Script Status Screen

To upload the test results for further analysis, do the following:

1. Click the checkbox before Upload results as.
2. Enter a name for your test results in the field Upload results as.
3. Click Upload.

After clicking Upload, a prompt screen is displayed. Click View Now and you are automatically taken to the *MyDeviceAnywhere* web site where you can view the results. Test results are shown under the Test Results tab in your *MyDeviceAnywhere* account.

Click Cancel when you want to dismiss the prompt screen and remain in Record & Playback.

When your script generates an Script Failure message, you might need to go back and redo the script. You might need to adjust the settings in SendKeys and WaitImage Properties screens. Anticipating network delay time and device response is an important part of designing a script that executes to completion.

5 Automation

DeviceAnywhere™ enables developers to perform in-network, on-device testing of mobile handsets and applications from the convenience of their desktops. They can have remote access to all aspects of handsets, including the buttons, LCD display, ringer, speaker, battery, power-plug, data-cable, and so on.

DeviceAnywhere Pro™ is an automation layer on *DeviceAnywhere*. *DeviceAnywhere Pro* supports a full-featured visual script editor. It offers an environment to create, schedule, and execute automated test scripts and Test Suites. These test scripts are used to perform functional and regression testing of mobile handsets and applications in wireless networks.

DeviceAnywhere Pro is available by clicking the Automation icon on the left in the Icon Bar.

5.1 Automation Overview

DeviceAnywhere Pro enables you to automate mobile application testing through the ability to do the following:

- Create test scripts. Save sets of commands that step through tests on one or more *DeviceAnywhere* handsets.
- Execute test scripts.
- Create and schedule Test Suites (groups of test scripts).
- Analyze test results.

In order to use *DeviceAnywhere Pro*, you must have both a *DeviceAnywhere* user account and a *DeviceAnywhere Pro* user license. If you do not have either of these, you should contact the administrator of your account, or you can contact us at Sales@DeviceAnywhere.com.

Before using *DeviceAnywhere Pro*, you should be familiar with *DeviceAnywhere* core functionality. If you are not familiar with *DeviceAnywhere*, read the Introduction and Chapters 1 to 3 of this User Guide to get acquainted with the *DeviceAnywhere* set up. These chapters explain how to use *DeviceAnywhere* to upload and test your mobile applications.

5.1.1 Automation View

Figure 5-1 illustrates the Automation view. It includes Scripts, Test Suites, Schedule, and the Dashboard.

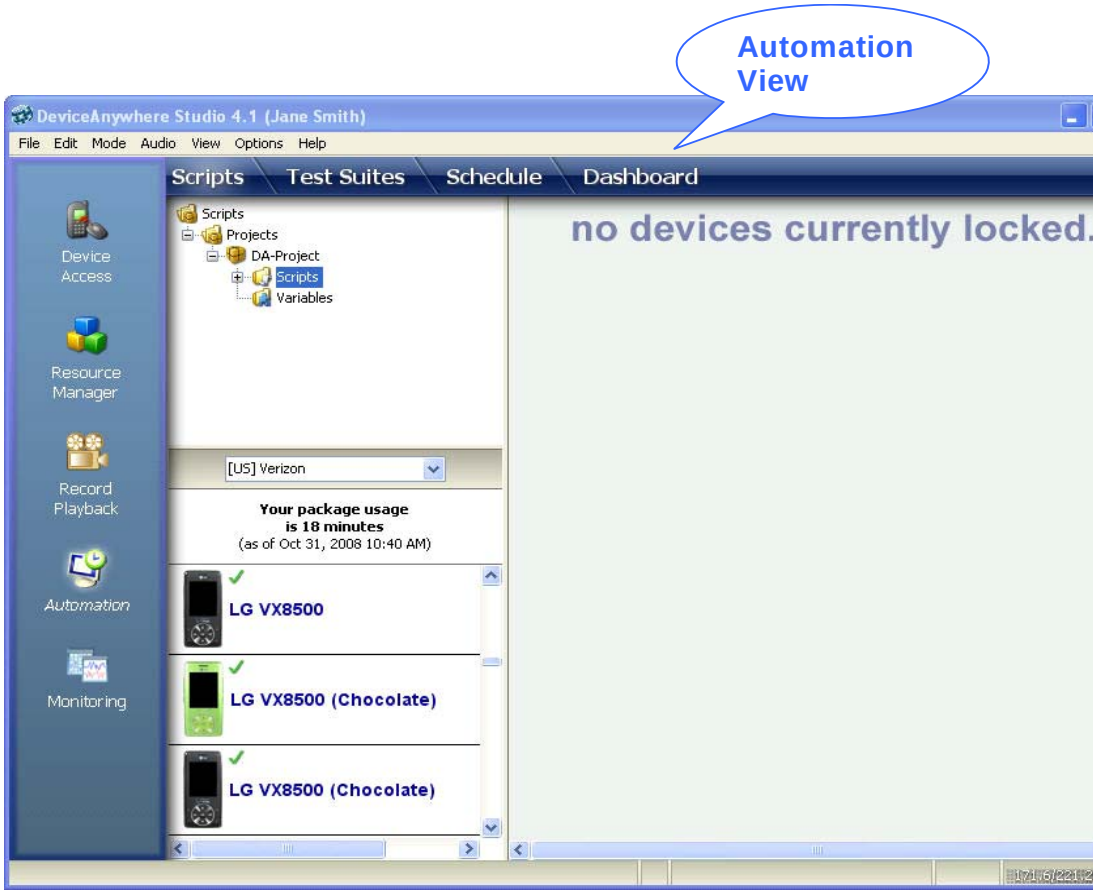


Figure 5-1 Automation View

A brief description of these tabs follows:

- Scripts—This area lets you create and organize scripts for different projects.
- Test Suites—Test Suites enables you to automate testing by defining a set of tests that can be run as a group.
- Schedule—The Schedule tab enables you to schedule Test Suites by date and time.
- Dashboard—The Dashboard enables you to view scripts as they run on the device.

DeviceAnywhere Pro is context sensitive. As you create folders and add scripts, the Menu Bar above the Automation view is expanded to include menus that support script building.

When a script is selected, the Menu Bar expands and displays the following menus:

- File—The file menu enables you to navigate directly to your *MyDeviceAnywhere* account. Enter code allows you to join a live device session. Use the Exit command to exit *DeviceAnywhere Studio*.
- Edit—The Edit menu supports cut and paste functions. *DeviceAnywhere Studio* also supports standard keyboard shortcuts, such as Control+C and Control+V.
- Mode—The Mode menu enables you to set the Key mode for an entry in the device. From this menu, you can select Alpha, Numeric, or Web modes.
- Audio—The Audio menu allows you to enable or disable the speaker and to change the size of the audio buffer.
- Script—The Script menu supports slot associations and global slot properties. Record and Playback controls are also available from this menu.
- Zoom—The Zoom menu enables you to select the grid size in the script editor.
- View—The View menu enables you to switch views in *DeviceAnywhere Studio*. It has the same function as the commands on the left Icon Bar.
- Options—The Options menu provides access to Error Management features that support *DeviceAnywhere Monitoring*. This feature is not active for *DeviceAnywhere Pro*. The Settings option enables you to set the Image cache for your system.
- Help—The Help menu provides access to the User Guide for this product and the latest Release Notes. It also supports many other links as described below.

5.2 Get Ready to Build a Test Script

5.2.1 Use a Project

All scripts are organized within projects. The first time you use *DeviceAnywhere Pro*, a default project called DA-Project is available to save scripts.

Only administrators are able to set up new Projects and assign Users. You should receive a Project assignment at the same time you receive a username and password to log into your *DeviceAnywhere Studio* account.

When you don't see your Project in the Explorer, contact your administrator. For more information on administrative tools, refer to "MyDeviceAnywhere" on page 279.

Figure 5-2 DA-Project

5.2.2 Create a Folder within a Project

Folders are used to organize scripts within a project.

How to Create a Folder:

1. Click on DA-Project to display the Scripts folder.
2. Right click on the Scripts folder. The File menu appears.
3. Select New Folder.
4. Enter the name of the folder. In this example, enter Automation.

Figure 5-3 illustrates the New Folder option in the drop-down menu.

Figure 5-3 New Folder Option

5.2.3 Create a Script

Now that you have created a folder to work in, it's time to create your first script.

How to Create a Script:

1. Right click on the folder. The File menu appears.
2. Select New Script.
3. Enter the name of the script. In this case, Send SMS. Script filenames must adhere to *DeviceAnywhere* file naming conventions. See Section 5.2.4 below.

Figure 5-4 shows the Send SMS script in the Explorer.

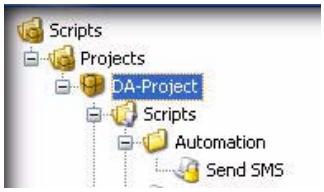


Figure 5-4 Send SMS Script

5.2.4 File Naming Conventions

DeviceAnywhere Studio has some restrictions on filenames that can be used. These guidelines apply to filenames for test scripts, monitoring scripts, and Macros. Follow the guidelines below in naming these files:

- Use A-Z upper and lower case letters followed by numbers.
- Numbers cannot be used at the beginning of filenames.
- Java keywords are not permissible.
- Do not use the generic name TestScript.

Figure 5-5 shows the empty Send SMS script in the script editor. You select a script to have it open in the script editor.

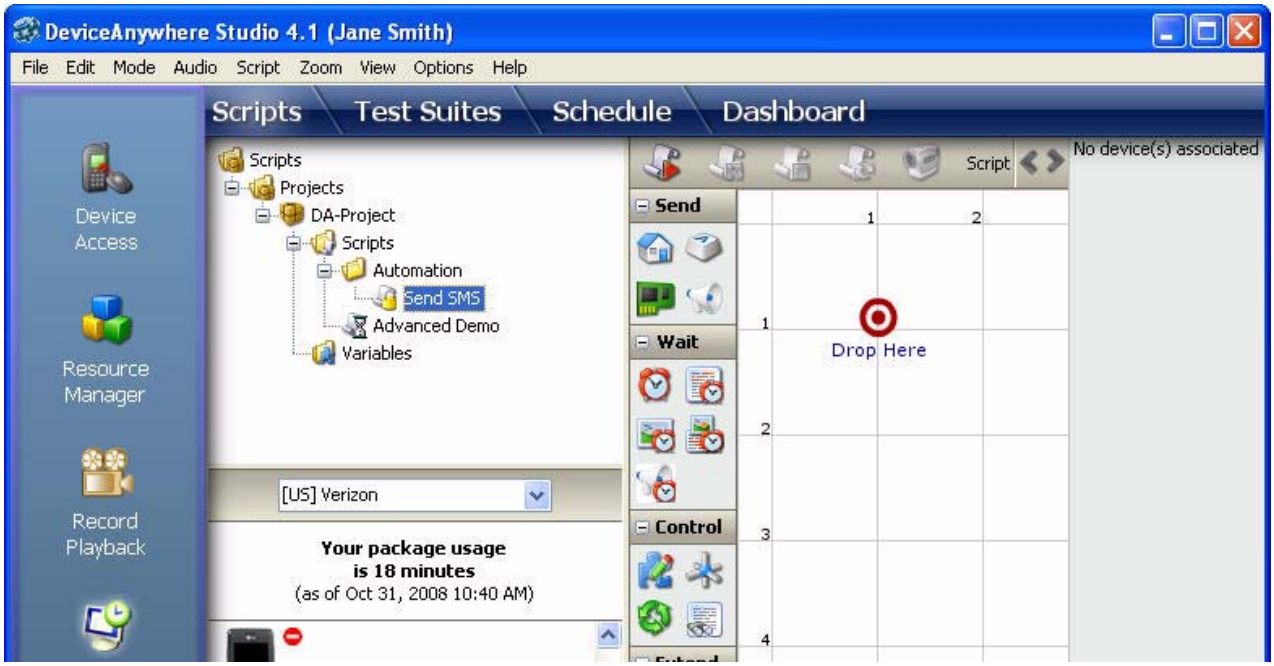


Figure 5-5 Empty Script in the Script Editor

One red target is shown at the top of the script. The red target indicates that your script is empty: no key presses have been recorded, and no script commands have been inserted into the script.

Generally, scripts are named by the test procedure that you are creating.

5.2.5 Add a Device to the Script

You must add a device to the script before recording a script. In Automation, more than one device can be added to a script at the same time. When you add a device, the device is displayed in the far right panel of the screen.

The procedure below adds a device to the script.

How to Add a Device to a Script:

1. Right click on a device in the Device list to add the device that you want to use.
2. Select Add to Script.

Figure 5-6 shows the Device menu with Add to Script selected.

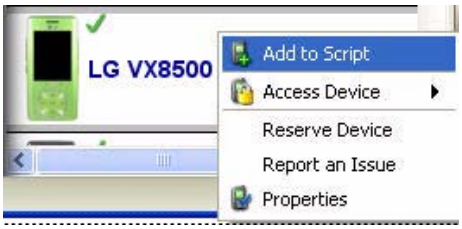


Figure 5-6 Add to Script

When you want to remove a device from the script, right click on the device. The Device menu is displayed. This time select Remove from script.

5.2.6 Access and Acquire a Device

Before you can create a test script, you must both add a device and acquire it.

How to Access and Acquire a Device:

3. Right click on the device in the workspace to display the Device menu.
4. Select Access Device.
5. Select Acquire Device.

Figure 5-7 shows how to select Access and Acquire Device commands on the Device menu.

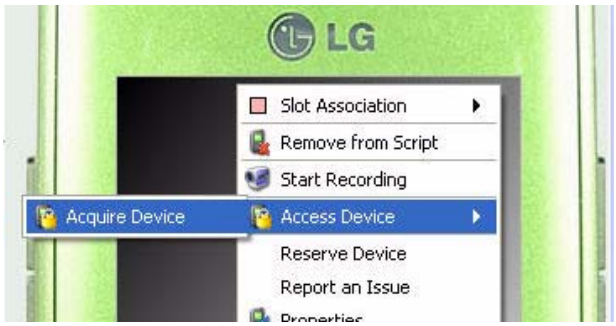


Figure 5-7 Access and Acquire a Device

Note: Before you start recording make sure that the device is awake and accepting input. Most devices go into a power save mode after a certain amount of inactivity. This amount of time varies per device.

5.3 Multi-Device Scripts

Each script must be associated with at least one device. Multiple devices can also be associated to a single script to allow you to do the following:

- Run the same script across similar devices
- Set up for tests that require more than one device (e.g., sending a text message from one device to another).

Device Bundles

Device bundles enable you to schedule a script to run across multiple sets of devices when using the automating scheduling in the Schedule tab. These device bundles are made from the available set of devices (i.e., those devices that have been added to the script). For detailed information on creating device bundles, refer to “Create a Device Bundle” on page 145.

Assigning Slots

Slot association is the way to associate script commands with devices in test scripts. If you assign a device to Slot 2, you assign a script command to Slot 2 if you want to execute on that device.

The first device that you add to a script is assigned to Slot 1 by default. All other devices must be assigned to a slot through the Slot association command in the Device menu. For example, if you want to send a message from one device to another, you can add two devices to a script and assign the devices to two different slots. Each command in the script is assigned a slot. Then, a single script can be run to send a message from one device to the other. For detailed information on slot association, refer to “Create a Device Bundle” on page 145.

Each slot is a different color to differentiate the devices. Up to 10 slots can be associated with one script (i.e., you can assign up to 10 devices to a script).

Global Allocation

TBD

5.4 Object-Oriented Scripting and Record and Playback

You can write scripts using multiple techniques that are supported by *DeviceAnywhere Pro*.

- One option is to use the icon-based scripting language. You use the visual script editor to drag and drop script commands into the script.
- Another option is to use the Record and Playback capabilities to record scripts by interacting with the device. As you press keys, script commands are automatically added to the script.

The best practice is to use a combination of both of these techniques. Use the Record and Playback toolbar in the script editor.

5.4.1 Adding Script Commands to the Script

All available script commands are located along the left side of the script editor.

You can add script commands to the script by dragging them from the left into the script editor.

Figure 5-8 shows a SendKeys command that has been dragged into the script editor.

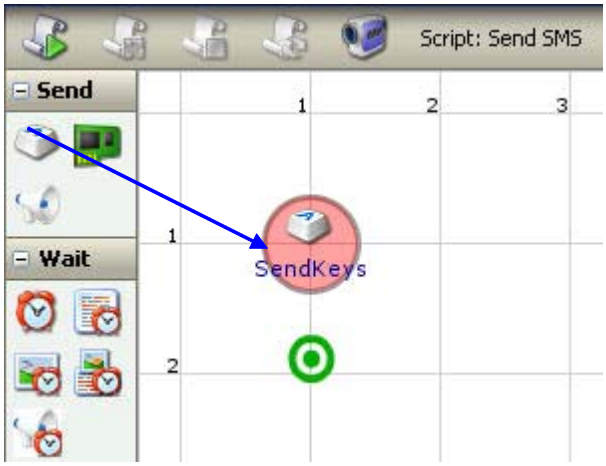


Figure 5-8 SendKeys Command Dragged to the Target

5.4.2 SendKeys Command Example

The SendKeys command can be used to execute key presses on a device. The SendKeys command will be used as an example to illustrate the properties and settings associated to each script command.

First look at the Text To Send field by opening the SendKeys command:

1. Right click on the SendKeys command. The Script Command menu appears.
2. Select Properties. Notice the Text To Send field is empty. This field stores the key presses that are made on the device or text that is manually entered to execute on the device.

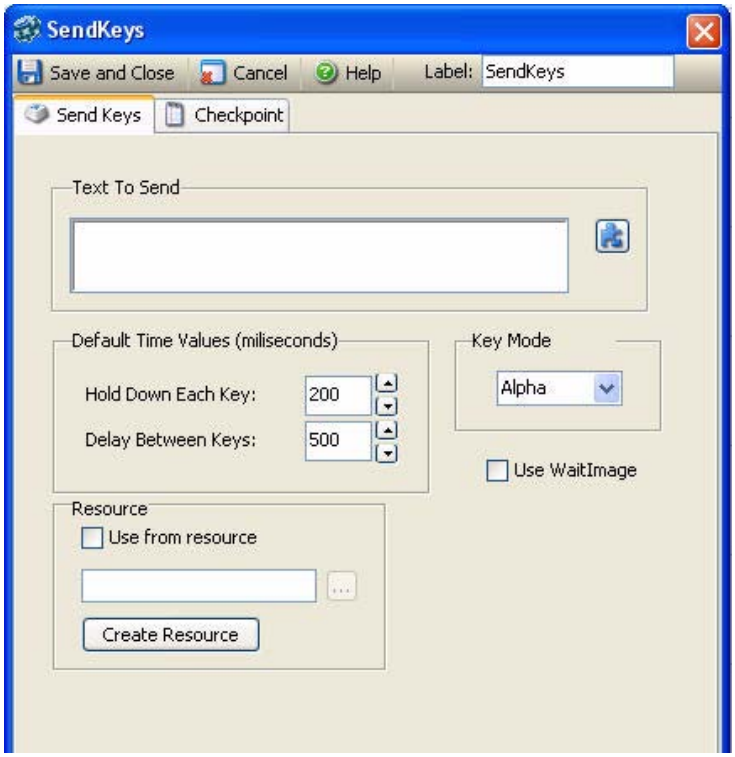


Figure 5-9 Empty Text To Send Field

5.4.3 SendKeys Properties

The Properties tab (default view) provides the following settings:

- Text to Send—The Text to Send field contains the key press associated to the SendKeys.
- Insert Variable—The Insert Variable icon (Puzzle piece) enables you to set Script variables.
- Default Time Values—The Hold Down Each Key field lets you specify the amount of time in milliseconds the key press is to be held down. The Delay Between Keys allows you to control the delay time between key presses.
- Key Mode—This field lets you select the key mode: Alpha, Numeric, or Web URL from the drop-down menu.
- Use WaitImage Checkbox—When this box is checked, two new tabs, WaitImage and Timeout, are automatically available in the Properties screen.
- This is a checkpoint step Checkbox—When the checkpoint checkbox is selected, the images that appear on the device screen before and after this step in script execution are captured.

5.4.4 WaitImage Tab in SendKeys Properties

Figure 5-10 shows the WaitImage tab in the SendKeys Properties screen. This tab is available when you check the Use WaitImage checkbox.

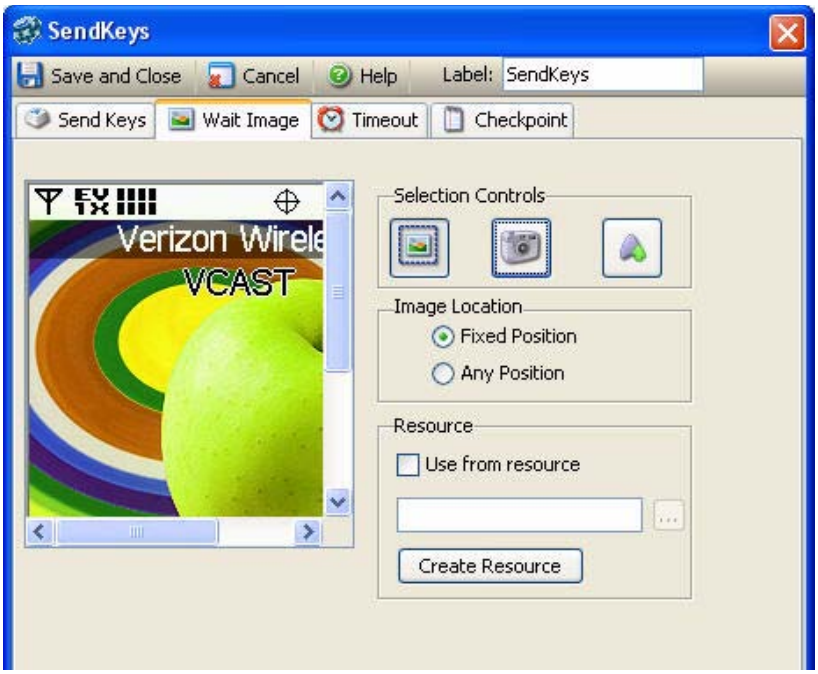


Figure 5-10 WaitImage Tab in SendKeys Properties

The WaitImage tab provides the following setting:

- Selection Control—This button is used to select a region of the reference image for the compare function. Click Selection Controls, then draw a rectangle on the image on the left to indicate the region that will be used in the compare.
- Camera—When the SendKeys is first added, the current device screen is captured. This reference image can be updated to the most current device screen by pressing the Camera button. If a new image is available, it will be displayed. You can pause the live recording, navigate to the correct frame by pressing the Device Video Controls, and return to live recording. This procedure allows you to enter any device screen that was previously recorded in the SendKeys command.
- Import—This option allows you to import an image from your desktop.
- Image Location—This indicator determines whether the compare function for the WaitImage command will search the image at a fixed position (exact coordinate location) or any position on the screen.
- Resource—These commands in the Resource window allow you to browse for an image in the Resource Manager or create a resource by storing the reference image from the SendKeys command. For more information on the Resource Manager, refer to “Resource Manager” on page 252.

5.4.5 Timeout Tab in SendKeys Properties

Figure 5-11 illustrates the Timeout tab for the SendKeys command when Use WaitImage is checked.

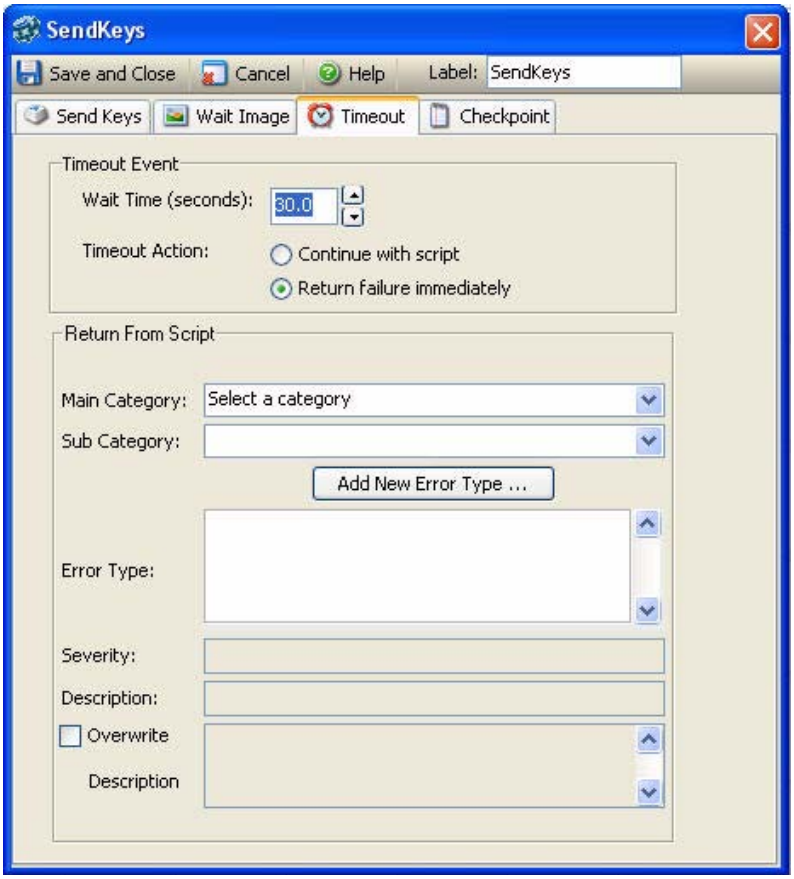


Figure 5-11 Timeout Tab

The Timeout tab shows two windows: Timeout Event and Return from script.

The Timeout Event window includes the following settings:

- Wait Time—Wait Time specifies how many seconds the command should wait for a defined action. For example while waiting for an image to download, you can add a timeout of 8 seconds. If the image does not load in 8 seconds, *DeviceAnywhere Studio* will declare a timeout.
- Time Out Action—Timeout Action enables you to specify the action to take in the event of a timeout. You can either (a) continue with the script or (b) return failure immediately.
  - In case of a timeout, the Continue with the Script option will not break the script execution. It will continue with the next command. For example, while waiting for an image to load, if the command experiences a timeout, you may choose to continue with the rest of the script and execute the remaining commands.
  - The Return Failure Immediately option enables you to stop script execution in case of a timeout. This may be required when the remaining script depends on the successful completion of this command. For example, if your script loads a WAP page, clicks on a link, and then performs tasks on the subsequent page, you can use the Return Failure Immediately option at the point where you click on the link. This terminates the script when the click on the link action times out.

The Return from Script window allows you to program error messages based on the custom Error Types, Categories, and Subcategories that you define for Error Management. Error Management provides granularity for error reporting and handling.

Five script commands support Error Management:

- SendKeys with WaitImage checked
- WaitImage
- WaitText
- WaitEvent
- WaitAudio
- Fail

Use the drop-down menus on the Timeout tab to select the Error Type, Category, and Subcategory to configure the WaitImage command.

For a detailed description of Error Management, refer to “Error Management” on page 166.

5.4.6 Enter Text in the Text to Send Field

There are two ways to add key presses to the Text To Send field.

- You can use recording capabilities.
- You can add text by entering it directly into the Text To Send field.

Recording the Key Presses

How to Record Key Presses and Store them in the SendKeys Command:

1. Right click on the SendKeys command.
2. Select Start Recording from the drop-down menu.
3. Press the End key four times on a device added to the script and acquired.
4. Right click on the SendKeys command.
5. Select Stop Recording.

Now take a look at the Properties of the SendKeys command. Double click directly on the Send-Keys command to open the Properties screen. The Text To Send field now contains the four Call End key presses you just recorded.

By right clicking on an individual SendKeys command and starting to record, you force all recorded key presses to be added to a particular SendKeys command. This is different than starting your recording from the toolbar, where the system places key presses in separate Send-Keys commands when there is too much hesitation between key presses. Recording into a single SendKeys command gives you more control over the organization of your test script.

Enter Text Manually in Text to Send Field

In this section, you copy a phone number from the Properties screen of a device, then paste it into the SendKeys Text to Send field.

How to Copy the Phone Number of a Device:

1. Right click on an active device in the workspace. The Properties screen is displayed.
2. Highlight the phone number from the phone number field.
3. Use the keyboard short cut Control+C to copy the phone number to the Clip board.

Now paste the phone number into the Text to Send field.

How to Type or Paste Text Directly into the Text To Send Field:

1. Drag a SendKeys command to the test script. Instead of recording your key presses, you are going to manually enter the phone number of the device.
2. Double click on the SendKeys command. The Properties screen is displayed.
3. Paste the phone number that was copied from the device into the Text to Send field. To paste the value, press Control+V. After pasting the phone number, remove any hyphens or spaces from the Text To Send field.
4. Make sure that the Key Mode is set to Numeric.
5. Click Save and Close.

5.4.7 Labeling Test Steps

All the labels on the script commands can be changed to make it easy to identify the actions in the test script.

How to Label Test Steps in a Script:

To change the label of a script command, double click the command to open the Properties screen. In the Label field (top right), enter a name that describes the test step. Click Save and Close.

5.4.8 Running a Script with Run from Here

The Run from Here command is an essential step in script building. *DeviceAnywhere Pro* enables you to execute a command at any point of a test script.

Use the Run from Here command in two ways:

- When you have typed or pasted text into a SendKeys, you must have the device execute the key strokes to get the device into the correct state. You have to set the device correctly before you can go on to the next step in the test script.
- When you are debugging, Run from Here is a great way to ensure that a long series of key presses runs correctly.

How to Select Run from Here:

1. Right click on the script command. The Script Command menu appears.
2. Select Run from Here.

The key presses entered into the Text To Send field are executed on the device.

5.5 Script Editor

The Script editor contains two toolbars and a workspace where you edit.

- The Script Command Toolbar, on the immediate left of the Script editor, contains the script commands that allow you to build a script.
- The Record and Playback Toolbar, above the Script editor, contains the controls to start or stop recording and to play or stop the execution of a script.
- The workspace is shown with a grid in which the script commands are placed. You can adjust the grid size using Zoom in the Menu Bar and selecting the zoom percentage.

5.5.1 Record and Playback Controls

Figure 5-12 illustrates the Play, Stop, Pause, Restart, and Record (left to right) buttons in the toolbar that control script execution.

Figure 5-12 Record and Playback Buttons

The Record and Playback toolbar has five controls (from left to right):

- Play Script—This control executes a script. Play Script is disabled when a script is already executing.
- Pause Script—This control enables pause of an execution of a script. Pause Script is only enabled during script execution.
- Stop Script—This control stops the execution of a script. Stop Script is only enabled during script execution.
- Restart Script—This control enables you to restart the execution from the start. Re-Run Script is available during script execution.
- Record Script— Click on Record Script, and a recording light blinks on the button indicating that recording has begun.

5.6 Working with Scripts

While working on a script, you may need to save changes, discard changes to start over, or export the script to your computer. The Script menu provides these options.

5.6.1 Script Status

Scripts are available for editing or saved in the database. A yellow lock on the script icon indicates that the script is open for editing. No other user has access to the script in the database. After you save a script, the yellow lock is removed. Refer to “How to Open a Script for Editing” below.

The Lock icon on the Send SMS script in Figure 5-13 indicates the script is locked for editing.

5.6.2 Open the Script Menu

How to Open the Script Menu:

To open the Script menu, right click on a script in the Explorer. The Script menu is displayed.

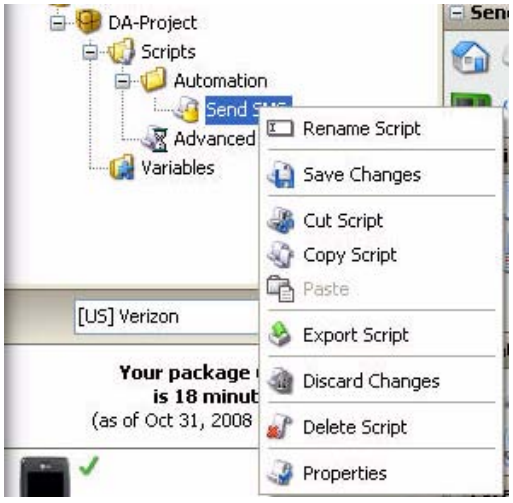


Figure 5-13 Script Menu

The Script menu has the following options.

- Rename Script—Rename Script lets you modify the name of a script.
- Save Changes—Save Changes saves the script to the database.
- Cut Script—Cut Script removes the script from the editor.
- Paste Script—Paste Script enables you to copy part of a script and paste it into a script.
- Export Script—The Export Scrip option enables you to export the script to your computer.
- Discard Changes—Discard Changes lets you get rid of all changes in the current script.
- Delete Script—Delete Script removes the script from the Explorer.
- Properties—This option opens the Script Properties screen. This screen provides a number of settings to view the history of scripts in a test environment. It also enables you to export and import data sets to be used in scripts. Refer to "Script Properties Screen" on page 93 for further details.

5.6.3 Perform Simple Tasks with the Script Menu

The following procedures describe how to use the Script menu to perform simple tasks.

Rename a Script

How to Rename a Script:

1. Right click on the script. The Script menu appears.
2. Select Rename Script. The file name is open for editing.

Save a Script

How to Save a Script to the Database:

1. Right click in the script editor. The Script menu appears.
2. Select Save Changes. The yellow lock is removed from the script icon in the Explorer.

Open a Script for Editing

How to Open a Script for Editing:

1. Right click on the script in the Explorer.
2. Select Lock/Edit Script. The yellow lock appears on the script icon in the Explorer.

Export a Script

How to Export a Script:

1. Right click on the script in the Script Editor.
2. Select Export. A Save file entry screen is displayed.
3. Navigate to the folder where you want the script saved.
4. Enter the name of the script.
5. Click Save.

The script is saved to your computer.

5.6.4 Script Properties Screen

The Script Properties screen contains a number of settings to control and review the history of scripts in a test environment. These tabs allow you to add comments about a script. Four tabs are available in this view.

How to Open the Script Properties Screen:

1. Right click on a script in the Explorer. The Script menu appears.
2. Select Properties.

Figure 5-14 illustrates the Script Properties screen.

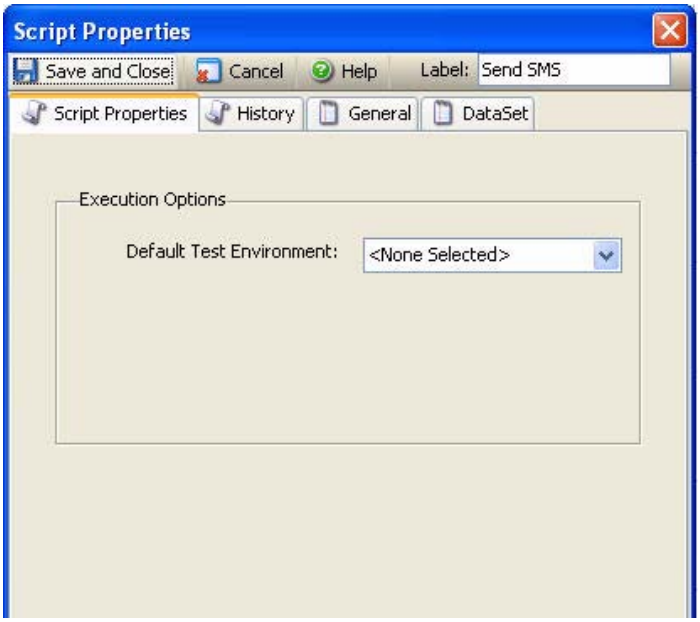


Figure 5-14 Script Properties Screen

5.6.5 Properties Screen Toolbar

The Properties screen toolbar at the top supports basic commands to save changes, cancel changes, and access online Help while working in the Properties screen. A brief description of these commands follows:

- Save and Close—This command lets you save the settings you entered into the Properties screen.
- Cancel—This command enables you to dismiss the Properties screen without saving any of the changes you have made.
- Help—This command provides access to online Help.
- Label—The Label entry field lets you enter the name for each SendKeys. This is a useful feature to label the steps in a long script.

All Properties screens in *DeviceAnywhere Studio* contain this toolbar and function in the same way.

5.6.6 Default Test Environment

DeviceAnywhere Studio supports scripting efforts for three test environments:

- Production
- Staging
- Development

When you first create a script, you may want to set the test environment that the script is written for. The test environments represent the stages of development. Test environment is used like a global parameter. When the test environment is moved from one stage to another, the scripts can be promoted or validated for a new test environment.

5.6.7 Script History

The History tab provides access to all versions of the script stored in the database. From here you can checkout a previous version or delete a version of a script. Once the database is changed, these changes become permanent.

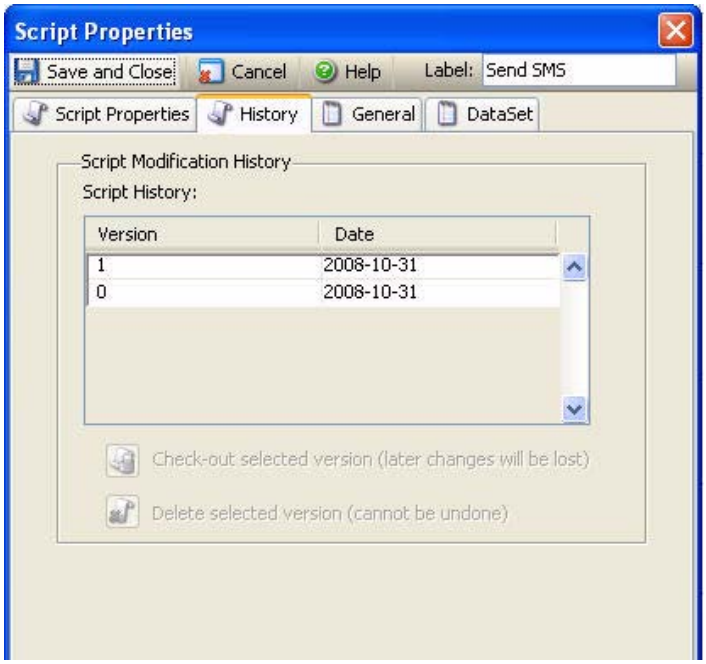


Figure 5-15 History Tab in Script Properties

5.6.8 Script File Properties

The General tab shows file properties such as creation date and the last date modified for a script. This tab also has an entry field to enter comments.

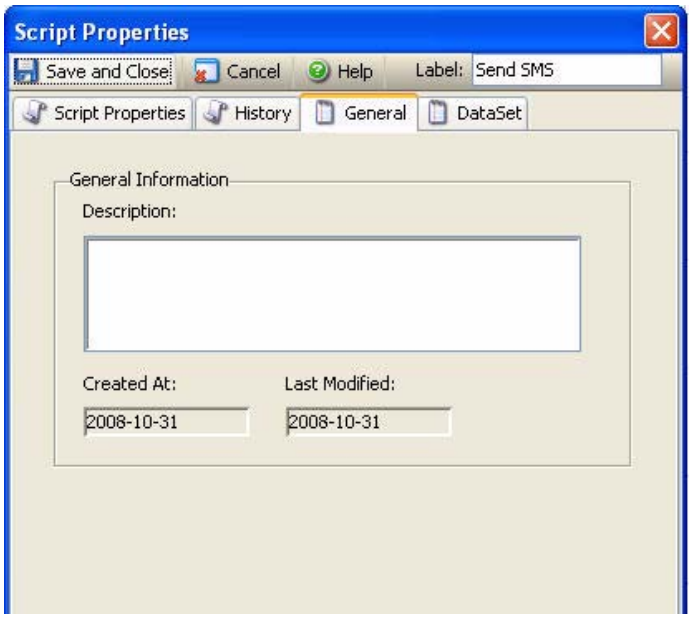


Figure 5-16 General Tab in Script Properties

5.6.9 Dataset

This tab is used to create datasets and import files in .csv format. Figure 5-17 shows a keyword (Message) sent to Weather.com shortcode that returns the weather forecast for a given zip code. Phone Num and Message are defined as variables in the DataSet tab. When the variables are evaluated during script execution, the keyword and short code are implemented in the Send SMS script. For detailed information on variables, refer to "Variable Features" on page 126.

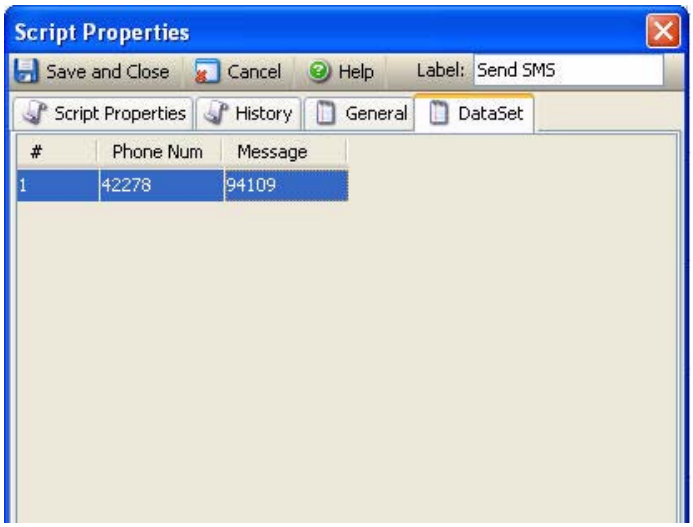


Figure 5-17 Dataset Tab in Script Properties

5.7 Recording a Test Script

5.7.1 Prepare for Recording

Before you begin recording your test script, you should run through your test at least once on the phone. You need to know what actions you want to record. This is important so that you don't get half-way through the recording and have to start over again.

If the recording is not exactly what you want, you can edit it. In most cases, you will need to edit recordings. However, the closer the recording is to your final script, the less editing you need to do later!

In this section, you create a test script by recording key presses. *DeviceAnywhere Pro* automatically inserts a SendKeys command into the test script. The test script that you create launches the WAP browser on the device and navigates to the carrier's home page.

Here is the test you will record. Try it out on the *DeviceAnywhere* phone of your choice.

1. Return to idle screen. The idle screen is the starting point for this test case and for most test cases that you eventually develop. It's important to make sure that the first thing you do in any test is to get the device to a known state. This way, each time you run the script, it doesn't matter where the device is in the device menu when acquired. The first thing your script should do is go back to the idle screen. The typical way to go back to the idle screen is to press the End button several times in succession so that no matter which device menu the device is currently in, you will end up on the idle screen. Usually four Call End key presses are sufficient.
2. Open WAP browser. Determine what the procedure is to launch the WAP browser. On most devices, you open the menu and select the Web icon.
3. Open carrier home page. Make sure you are on the carrier home page. This is important because when you are creating your script, you want it to go directly to the carrier home deck. If the device is on another wap page, you can return it to the carrier home page from the options in the browser.
4. Return to idle screen.

5.7.2 Recording the Test Script

Now that you are familiar with the test you want to record, you are ready to record your test script.

Table 5-1describes the key presses to record in this end-to-end script.

Table 5-1 WAP Browser Test Script

Action	Step	Expected Result
1. Return to idle	Press the End key four times.	Device returns to idle screen.
2. Open WAP browser	Go to the main menu and select the Web icon.	Device returns a web menu screen or goes directly to the carrier home page.
3. Open carrier home page	For most devices, you are automatically taken to the carrier home page. Other devices provide an intermediate screen where you can select the carrier home page.	Carrier home page opens successfully.
4. Return to idle	Press the End key.	Device returns to idle screen.

How to Record a Script:

1. Start by clicking the Record icon in the toolbar.
2. Operate the device by following the steps outlined in Table 5-1. Now that you are recording, as you press keys, you see the SendKeys commands automatically added to the script editor.
3. Click the Record icon to stop recording your key presses.

Note: Before you start recording make sure that the device is awake and accepting input. Most devices go into a power save mode after a certain amount of inactivity. This amount of time varies per device.

DeviceAnywhere Pro is context sensitive. In this case, clicking the Record Icon in the toolbar a second time stops the recording.

Once you have completed the recording, your test script should look similar to the script in Figure 5-18.

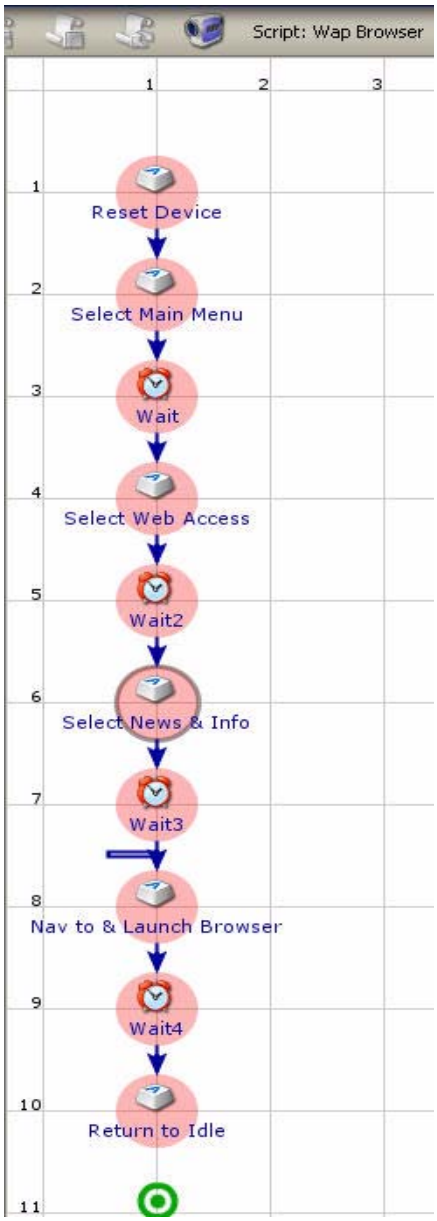


Figure 5-18 WAP Browser Test Script

Each pink circle in the script indicates a separate command.

- Key presses are recorded into a single SendKeys command when there are no pauses. For instance, when you press [End][End][End][End] with little time between each key press, all four of these key presses are included in one SendKeys command.
- If you pause between actions while recording, a Wait command is inserted. The Wait command includes the amount of time the user paused during script recording. If the Wait was unintentional or if it is not the length of time desired, you are able to go back and edit the Wait length.

5.7.3 Saving Changes

DeviceAnywhere Studio has version control built into the product. Scripts are automatically saved to the database as you are working. However, Saving Changes writes an official copy of the script to the database for the script History. When you write the script to the database, the Lock icon is removed from the script to indicate that it is released for editing. Any user in the Project can access the script at this point.

After saving a scripts, right click the script in the Explorer and select the Lock/Edit command to edit the script further after saving changes.

Figure 5-19 below shows you how to save script changes.

Figure 5-19 Saving Script Changes

5.7.4 Playing the Test Script

Now that you have created your first test script, it is time to play your script. It is important to make sure that each individual test script plays back successfully at least one time before you move on to building your next test script.

To play the test script, do the following:

1. Select your script in the Explorer to the left.
2. Click the Play button in the toolbar.

When the script begins to execute, the green Play button changes from green to gray. The Red stop button now becomes active while the script is executing, as well as the Pause and Restart buttons.

For now, let the script play to completion. As the script executes, notice that the script command that is currently executing has a dark grey circle around it. This feature helps you track the execution of your script.

When the script finishes, a dialog box pops up and displays the status of the execution.

5.7.5 Uploading Test Results

This section describes how to view the results of a test script that was executed.

- Upload the results to your *MyDeviceAnywhere* account
- View the results with images

The Script Status screen enables you to select options to upload test results.

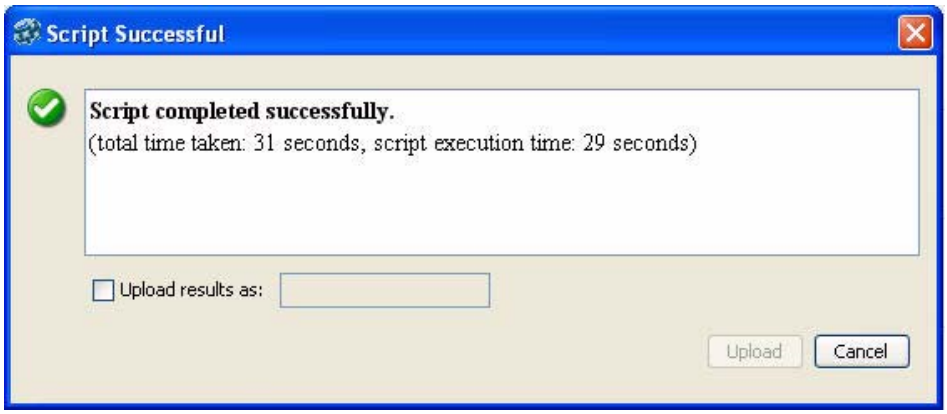


Figure 5-20 Script Status Screen

How To Upload the Test Results:

1. Click the checkbox before Upload results as.
2. Enter a name for your test results in the field Upload results as.
3. Click Upload.

After clicking Upload, a prompt screen is displayed. Click View Now and you are automatically taken to the *MyDeviceAnywhere* website where you can view the results. Test results are shown under the Test Results tab in your *MyDeviceAnywhere* account.

Click Cancel when you want to dismiss the prompt screen and remain in *DeviceAnywhere Pro*.

Figure 5-21 shows the test results in *MyDeviceAnywhere*.

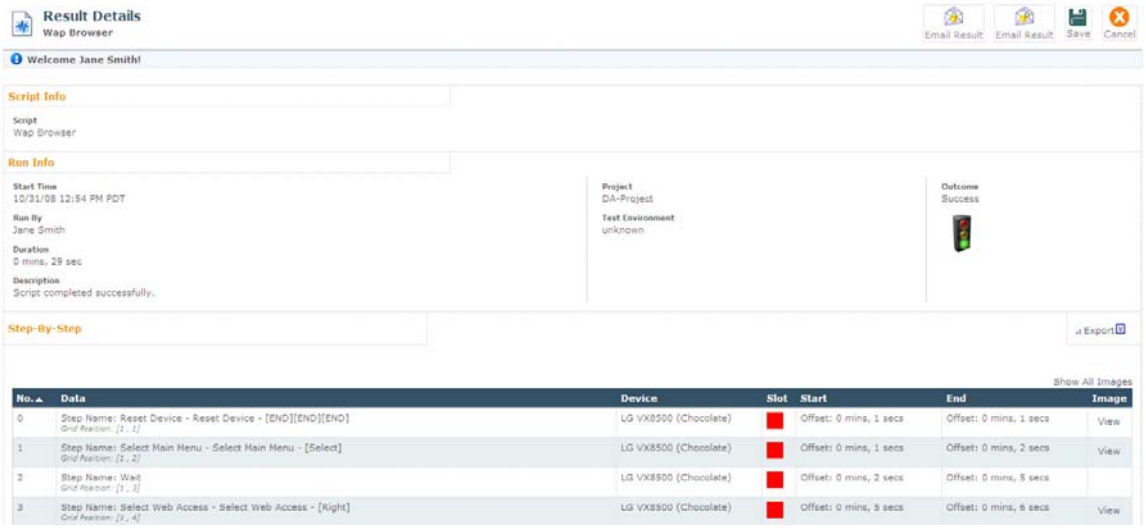


Figure 5-21 Wap Browser Test Results

Figure 5-21 shows the test results for the Wap Browser script. Each step is numbered sequentially. The labels that you entered for each SendKeys and WaitImage command are shown in the report.

While the script was executing, the screen shots were captured from the device screen before and after each command executed within the script. These screen shots are available in the Test Results.

By default, the results don't display the images. If you click the Show All Images link, all the images are displayed. If you don't want to show all the images, click the View link only at the step where you do want to see the image.

5.8 Building a Send SMS Script

So far, you've learned to create a simple test script using the Record command in the toolbar. In this section, you use script commands to build a Send SMS script in the script editor. The Send SMS script has two Macro parameters or local variables defined in the script to make it generic for any SMS transactions.

In this example, Send SMS is sending a keyword to the weather.com shortcode 44782. However, the Send SMS script could just as easily be sending an SMS message to another device or service. This section introduces you to best practices in creating test scripts that can run against 30 devices or more.

Two terms, Test Procedures and Implementations, are used to describe the ideal structure of a test script when it is used across multiple devices.

> Test Procedure—A Test Procedure (main script) is a high-level script comprised of individual test steps or actions.
>
> Implementation—An implementation is a low-level script that is device specific. It executes the commands on a device (e.g. key presses) to accomplish the actions of a test step.

This concept best supports the development of multi-device scripts.

Understanding Resources is an essential part of script creation. Refer to the "Resource Manager" on page 252 for a detailed description of Resources.

In this example, three devices are used to create a multi-device Send SMS script.

- LG VX8500
- BlackBerry 8100 Pearl
- Samsung A707

> Note: The LG VX8500 Chocolate is programmed to a short wait time before it becomes inactive, and the input keys are locked. Press the Shift key on your keyboard and the End key on the device at the same time to activate the device from Sleep mode. Or use the Wake Up Device or Reset Device options in the Device menu to activate devices that have entered Sleep mode

However, you can add any other devices as needed by using developing Macro resources. Each device has a unique series of key presses to send an SMS. Therefore, take into consideration the device you are using and create your scripts accordingly.

Figure 5-22 shows the main script for sending an SMS. It is a visual representation of a test procedure that applies to any device. The test procedure is created with four Macro calls dragged into the script editor.

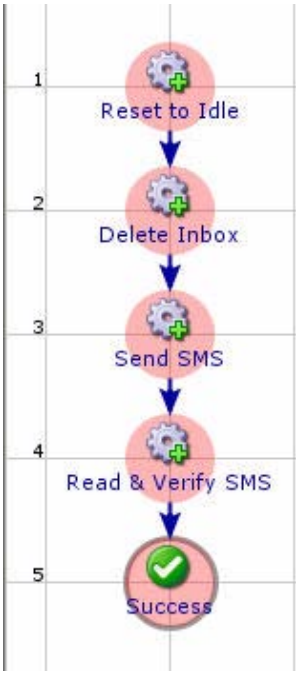


Figure 5-22 Send SMS Script

Follow the procedures below to create the Send SMS main script.

5.8.1 Create the Send SMS Test Procedure

Create New Script and Acquire a Device

How to Create a New Script:

1. In the Automation view, right click on a folder.
2. Select New Script.
3. Enter the name of the new script. In this case, Send SMS.
4. Right click in the Device list.
5. Select the LG VX8500.
6. Select Add to Script.
7. Right click on the LG VX8500 to display the Acquire Device menu.
8. Select Access Device.
9. Select Acquire Device.

5.8.2 Create the Test Procedure

Create the main script by dragging four Macro calls into the script.

How to Create a Main Script:

1. Drag a Macro call from the left of the script editor into the script.
2. Double click the Macro script command. The Properties screen is displayed.
3. Enter Reset to Idle in the Label entry field.
4. Click Save and Close.
5. Continue dragging three more Macro calls into the script.
6. Label each Macro call in the following order: Delete Inbox, Send SMS, Read and Verify SMS.
7. Drag a Success script command to finish the script.

A Success script command is optional.

Figure 5-23 illustrates a Macro call being dragged into the script.

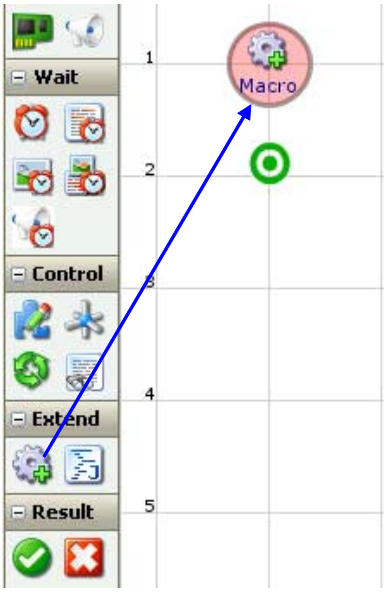


Figure 5-23 Macro Call in Main Script

5.8.3 Create the Macro Resources

Now that the main script is ready, you need to create resources in the Resource Manager for the Send SMS script. Click on Resource Manager in the Icon Bar to start creating resources.

Set up a Macro Resource Library

In this section, you create a Resource library, a library folder, and resources for the Send SMS test script.

1. In the Resource Manager, right click Resources in the Explorer. The Resource menu appears.
2. Select Add Resource Library. A new Resource library appears in the Explorer.
3. Enter the name of the Resource Library. In this case, enter My Resource Library.
4. Right click the My Resource library. The drop-down menu appears.
5. Select Add Category. This action creates an empty Send SMS folder.

All the Macro resources that are used in the Send SMS test script are stored in this folder.

Create Macro Resources

In this section, you create two Macro resources—Reset to Idle and Send SMS.

- Reset to Idle is the first Macro in the script and provides a very straightforward example of script building.
- Send SMS is an interesting script, since it shows the use of Macro parameters that are evaluated when Send SMS is executed.

The following procedures are used over and over to create a Macro resource and Macro instances in a Test Procedure.

To complete the entire Send SMS script, you'll need to create a Macro resource for the other Macros (Delete Inbox and Read & Verify SMS) in the Test Procedure.

How to Create a Macro Resource:

1. Right click the Send SMS folder.
2. Select Create Resource Type.
3. Select Macro. A new Macro resource appears in the Explorer.
4. Enter Reset to Idle. This name corresponds to the first Macro call in the Send SMS test script.
5. Create three more Macro resources by following Step 1 through Step 4. In this case, enter the following names: Delete Inbox, Send SMS, and Read & Verify SMS, respectively.

Figure 5-24 shows the Macro resources in the Send SMS folder.

Figure 5-24 Macro Resources for Send SMS

5.8.4 Create Macro Instances—Reset to Idle

In this example, you create three Macro instances for Reset to Idle. Three devices will be implemented:

- LG VX8500
- BlackBerry 8100 Pearl
- Samsung A707

Reset to Idle—LG VX8500

The first device used is the LG VX8500.

How to Create a Macro instance:

1. Select a Macro in the Explorer. In this case, select Reset to Idle.
2. Right click the LG VX8500 device in the Device list.
3. Select Create Resource Instance. A new Macro instance appears in the Resource Instance window.

Double click on the Resource Instance. An empty Macro instance appears in the script editor.

Figure 5-25 shows an empty script. Macro instances are created just like a test script by dragging script commands into the editor.

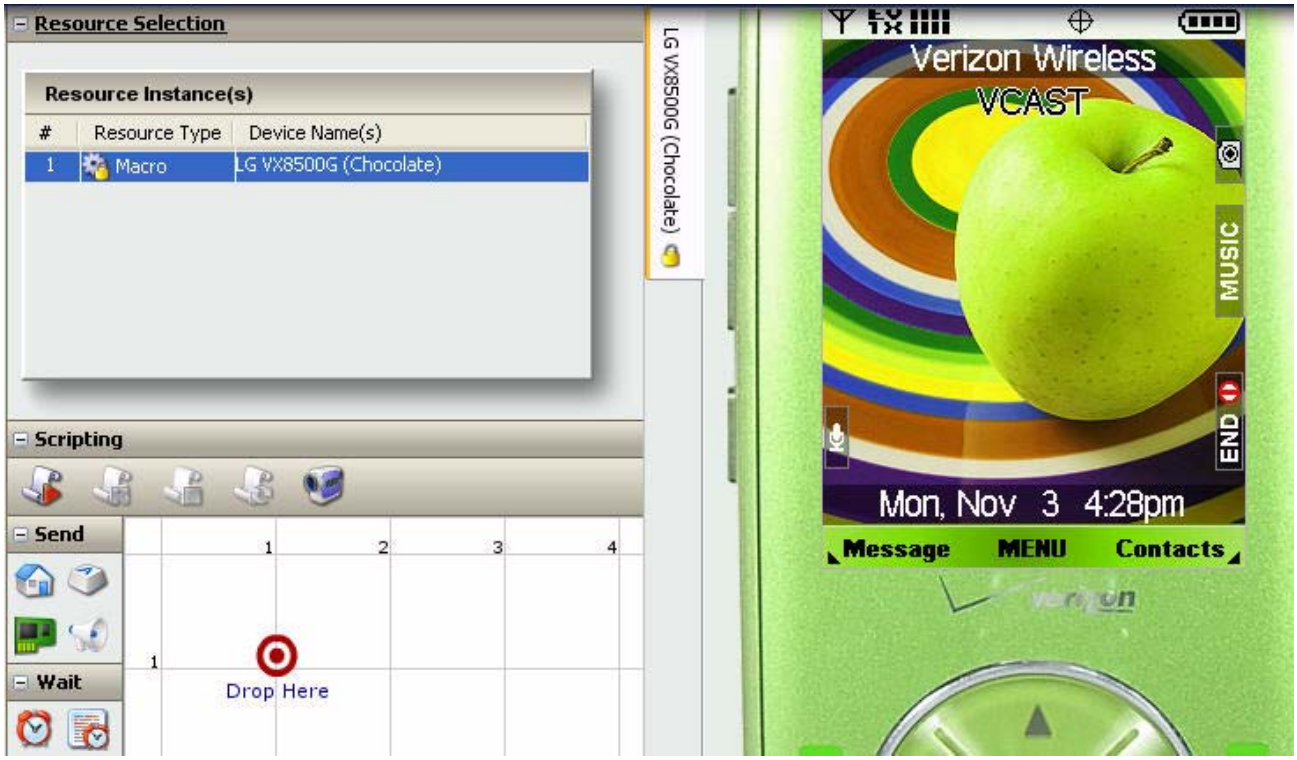


Figure 5-25 Empty Macro Resource—LG VX8500

In creating a Macro instance, device behavior has to be taken into consideration. Initialization of a device is critical to running test scripts successfully. The test script must be robust enough to activate a device no matter what state it's in.

Figure 5-26 shows the Reset to Idle Macro instance for the LG VX8500.

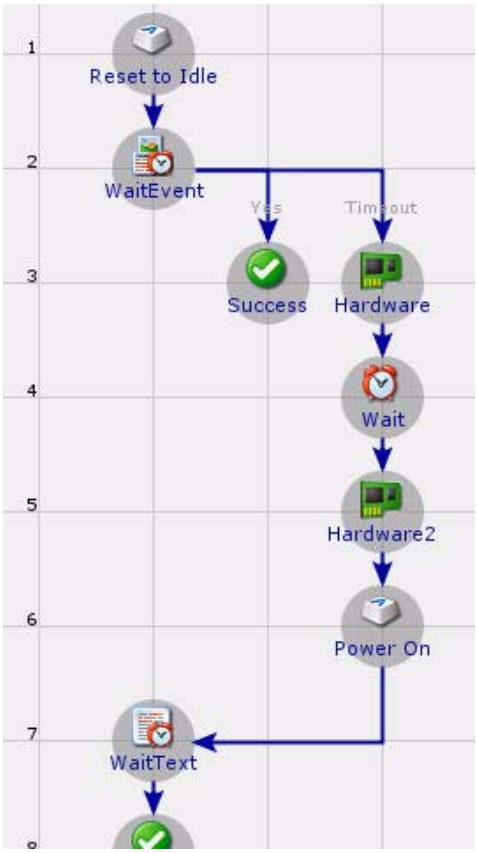


Figure 5-26 Reset to Idle—LG VX8500

This Macro instance for the LG VX8500 is designed with two initialization paths.

- The first path is a standard initialization. It consists of pressing the End key five times. In general, these key presses return most devices to the idle state, regardless of their positions within the device menus. This first path is called Yes in Reset to Idle.
- The second path is a robust initialization. It executes steps to activate the device from Sleep mode and unlock the keypad. In this case, the initialization includes disconnecting and reconnecting the battery, turning on the device, and checking for the idle screen. The second path is called Timeout, since this path is executed when the device fails to display the idle screen. This is where script execution branches to a robust initialization sequence.

The key presses and checkpoints that are executed for the LG VX8500 are summarized in Table 5-2 that follows.

Reset to Idle—BlackBerry 8100 Pearl

Now, add a Macro instance for the BlackBerry 8100 Pearl.

1. Select Reset to Idle in the Explorer.
2. Right click the BlackBerry 8100 Pearl in the Device list.
3. Select Create Resource Instance. A new Macro instance appears in the Resource Instance window.

Double click on the Resource Instance. An empty script for the BlackBerry 8100 Pearl appears in the script editor.

Figure 5-27 shows another Macro instance added to the Resource Instance window. This time the Macro instance for the BlackBerry 8100 Pearl is listed.

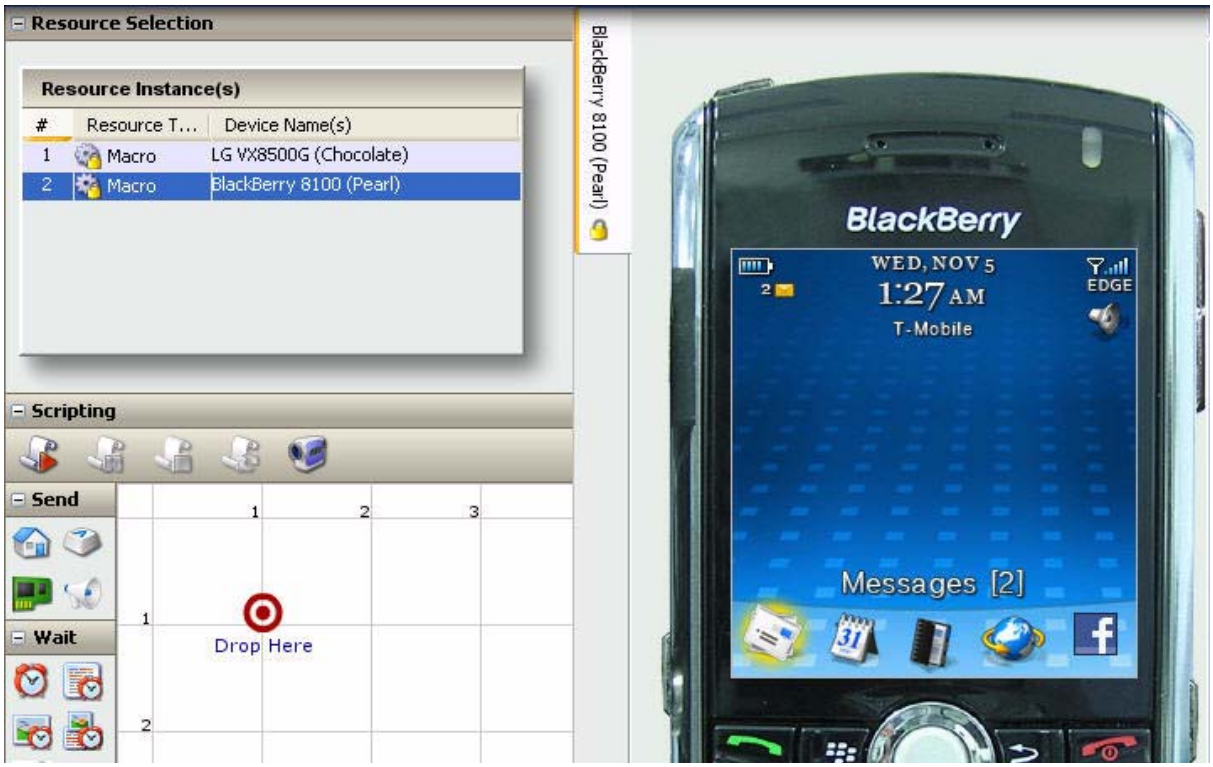


Figure 5-27 Reset to Idle—BlackBerry 8100 Pearl

The BlackBerry 8100 Pearl is a smartphone.

Figure 5-28 shows the finished implementation for the BlackBerry 8100 Pearl. It shows the same two initialization paths as the LG VX8500.

Figure 5-28 Reset to Idle (Macro Instance—BlackBerry 8100 Pearl)

The key presses and checkpoints that are executed for the BlackBerry 8100 Pearl are summarized in Table 5-2 that follows.

Reset to Idle—Samsung A707

Now, add a Macro instance for the Samsung A707.

1. Select Reset to Idle in the Explorer.
2. Right click the Samsung A707 in the Device list.
3. Select Create Resource Instance. A new Macro instance appears in the Resource Instance window.

Double click on the Resource Instance. An empty script for the Samsung A707 appears in the script editor. Create the Macro instance by using the icon-based scripting language or using the Record and Playback capabilities on the toolbar.

Figure 5-29 shows another Macro instance added to the Resource Instance window. This time the Macro instance is for the Samsung A707.

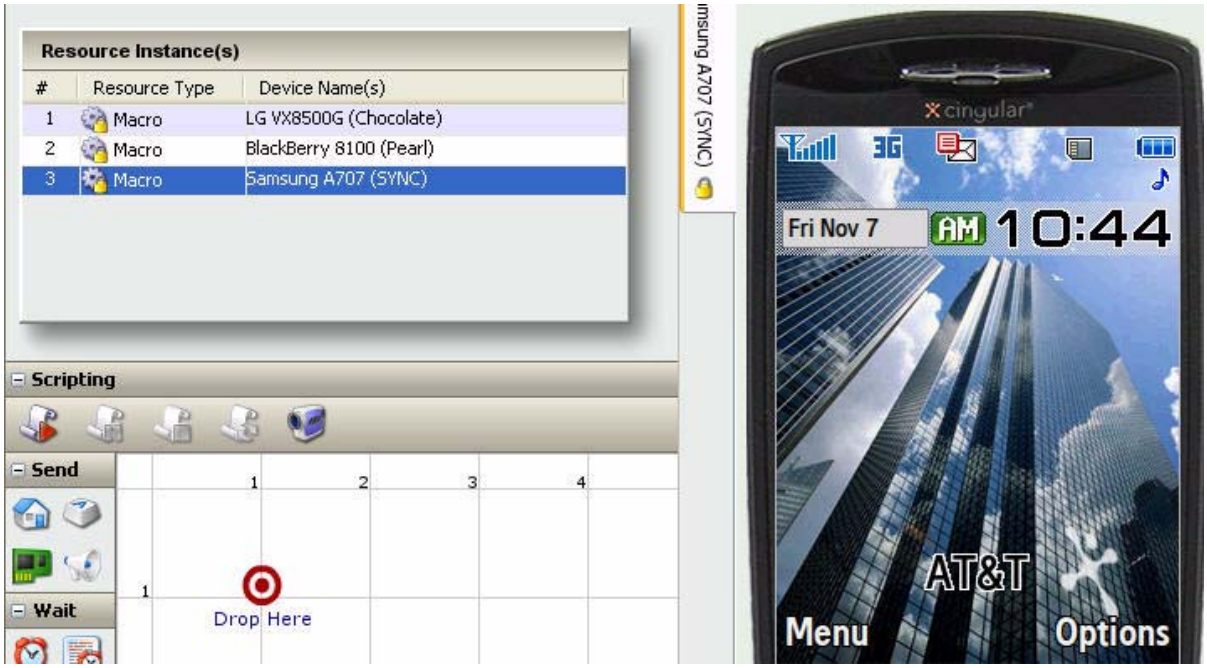


Figure 5-29 Empty Macro Instance—Samsung A707

Figure 5-30 shows the finished implementation for the Samsung A707. It shows the same two initialization paths as the LG VX8500 and the BlackBerry 8100 Pearl.

Figure 5-30 Reset to Idle (Macro Instance—Samsung A707)

The key presses and checkpoints that are executed for the Samsung A707 are summarized in Table 5-2 in the following section.

5.8.5 Reset to Idle—Macro Instance Summary

Table 5-2 provides a summary of the key presses, checkpoints, and branching logic that are used in the low-level scripts for the LG VX8500, BlackBerry 8100 Pearl, and the Samsung A707, respectively.

In a comparison of these scripts, the design of Reset to Idle for all three devices is the same. They all rely on two initialization paths: one for a quick and easy reset, and the other for a device in an unknown state.

The key presses that are stored in the Text To Send field and executed on each device are different; they correspond to the device menu hierarchy and the menu options specific to each device.

For example, notice the key presses in Table 5-2 for the BlackBerry 8100 Pearl in Step 1. The BlackBerry is known for nested menus. Step 1 executes a series of Back commands to force the BlackBerry device to return to the Idle screen when it is stuck in nested menus. The other devices do not require this string of commands to return to the Idle state.

Table 5-2 Reset to Idle

Action	Expected Result	LG VX8500	BlackBerry 8100	Samsung A707
1. Reset to Idle	Device returns to the Idle screen.	[Power][Power] [Power][Power]	[Back][Back] [Back][Back] [Back][Back] [Back][Back] [Power][Power]	[Power][Power] [Power][Power]
2. Verify Idle (WaitEvent)	The current device screen is compared to the reference image of the Idle screen. Two branches are possible: 1) the current device screen matches the Idle screen; 2) the current device screen does not match the Idle screen. In this case, the script executes the Timeout path.	WaitImage compares the current device screen to the Idle screen. Two branches are possible: 1) Yes—initialization succeeds; 2) Timeout—additional commands are executed to reset the device.	WaitImage compares the current device screen to the Idle screen. Two branches are possible: 1) Idle Image—initialization succeeds; 2) Timeout—additional commands are executed to reset the device.	WaitImage compares the current device screen to the Idle screen. Two branches are possible: 1) Idle Image—initialization succeeds; 2) Timeout—additional commands are executed to reset the device.
3. Verify Idle A. Success	Reset to Idle script completed successfully.	Reset to Idle script completed successfully.	Reset to Idle script completed successfully.	Reset to Idle script completed successfully.
3. Verify Idle B. Timeout Hardware disconnect	Reset to Idle script failed. Begin reset by disconnecting the battery.	Hardware Disconnect battery.	Hardware Disconnect battery.	Hardware Disconnect battery.
4. Wait	Wait 3 seconds for device shut down.	Wait 3 seconds for device shut down.	Wait 3 seconds for device shut down.	Wait 3 seconds for device shut down.
5. Hardware Reconnect	Reconnect the battery.	Reconnect the battery.	Reconnect the battery.	Reconnect the battery.

Table 5-2 Reset to Idle (Continued)

Action	Expected Result	LG VX8500	BlackBerry 8100	Samsung A707
6. Power On	Power on the device by pressing the Power key.	[Power]	[Power]	[Power]
7. Verify Idle WaitText or WaitImage	Verify device has returned to the Idle screen.	WaitText compares the current device screen to the carrier Idle screen. This command focuses on test only and will not fail when the carrier's idle screen is updated with a new color or new image.	WaitImage compares the current device screen to the carrier Idle screen. This command focuses on the image and fails when the carrier's idle screen is updated with a new color or new image.	WaitImage compares the current device screen to the carrier Idle screen. This command focuses on the image and fails when the carrier's idle screen is updated with a new color or new image.
8. Success	Reset to Idle script completed successfully.	Reset to Idle script completed successfully.	Reset to Idle script completed successfully.	Reset to Idle script completed successfully.

Key Points:

- To support multi-device scripts, create a Test Procedure with Macro calls to define the actions of the test.
- Implement low-level scripts that are device specific in the Resource Manager.
- Take into consideration device behavior as you design the Macro instance.

Reset to Idle is the first Macro resource in the Send SMS Test Procedure. You'll need to create Delete Inbox and Read & Verify SMS Macro resources with the Samsung A707, or develop one implementation for one device consistently, before moving on to the next section.

5.9 Create Macro Instances—Send SMS

In this section, you create a Macro resource called Send SMS. The Send SMS implementation has two Macro parameters (local variables) defined in the script to make it generic for any SMS transactions.

In this example, Send SMS is sending a keyword to the weather.com shortcode 44782 to obtain a weather forecast for a given zip code. However, the Send SMS script could just as easily be sending an SMS message to another device or service.

Only the Macro instance for the Samsung A707 is presented. However, since the main script is composed of generic 'actions,' any device can be implemented by adding other Macro instances for specific devices.

Figure 5-22 shows a visual representation of the steps needed to send a text message on the Samsung A707.

Figure 5-31 Send SMS—Samsung A707 Macro Instance (Top)

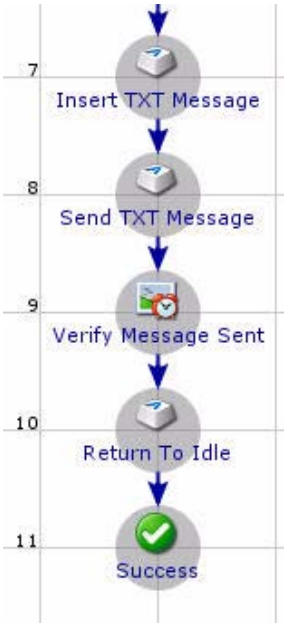


Figure 5-32 Send SMS—Samsung A707 Macro Instance (Bottom)

Table 5-3 provides a summary of the key presses and checkpoints that are used in the low-level script for the Samsung A707.

Table 5-3 Send SMS—Samsung A707

Action	Expected Result	Samsung A707
1. Return to Idle	Device returns to the Idle screen.	[Power][Power] [Power][Power] [Power]
2. Verify Idle (WaitImage)	The current device screen is compared to a reference image of the Idle screen.	WaitImage succeeds.
3. Nav to Messaging	Device navigates to the main Messaging screen.	[SoftL][Up][Select]
4. Verify Messaging	The current device screen is compared to the main Messaging screen.	WaitImage succeeds.
5. Select New Message	The New Message option is selected from the menu.	[Select]
6. Insert phone number	A variable named Num is entered into the phone number field.	%Num%

Table 5-3 Send SMS—Samsung A707 (Continued)

Action	Expected Result	Samsung A707
7. Insert TXT message	A variable named Msg is entered into the message body.	[Down]%Msg%
8. Send TXT message	Send Message option selected.	[Select][SoftR] [SoftR][Select]
9. Verify Message sent (WaitImage)	The current device screen is compared to the Message sent screen and succeeds.	WaitImage succeeds.
10. Return to Idle	Device returns to the Idle screen.	[Power][Power] [Power]
12. Success	Script runs to completion successfully.	Script succeeds.

The following sections provide a step-by-step description of how to build the Send SMS Macro resource for the Samsung A707.

5.9.1 Return to Idle

To record key presses that reset the device, do the following:

1. Drag the SendKeys command into the script editor.
2. Right click on the SendKeys command.
3. Select Start Recording.
4. Now press the Power key many times on the Samsung A707 on the right side of the script editor.
5. Right click on the SendKeys command.
6. Select Stop Recording.
7. Change the label of the SendKeys command by entering Return to Idle in the Label field.
8. Click Save and Close.

Figure 5-33 Samsung A707 Power key

5.9.2 Verify Idle

Next you need to verify that the Samsung A707 is on the idle screen. In this case, you use a WaitImage command.

How to Add a WaitImage Command and Capture a Reference Image:

1. Drag the WaitImage command to the green target.
2. Double click the WaitImage command to view the Properties.
3. Use your mouse and select an area of the screen to create your reference image.
4. Change the label of the WaitImage command by entering Verify Idle in the Label field.
5. Click Save and Close.

Figure 5-34 shows the Samsung A707 idle screen. When the Macro instance is executed, this reference image is compared to the image returned from the device in real-time to verify that the device is in the idle screen.

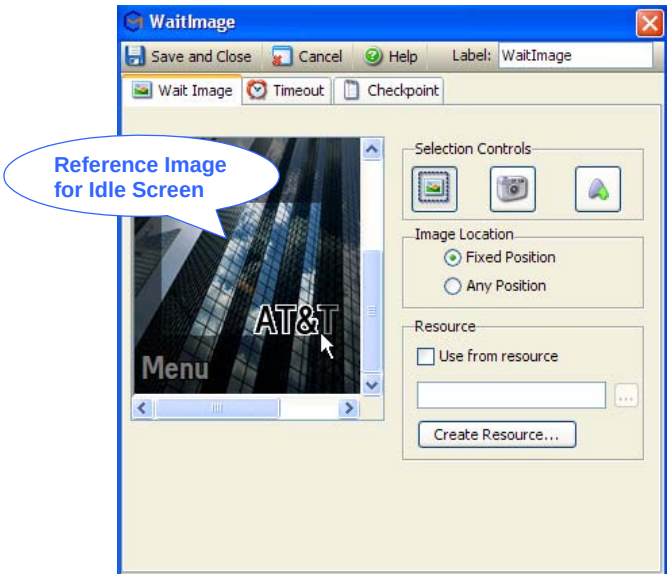


Figure 5-34 Samsung A707 Idle Screen

5.9.3 Nav to Messaging

This time you record key presses to navigate to the Messaging screen on the Samsung A707.

Record Key Presses to Navigate to the Message Screen:

1. Drag a SendKeys command to the green target.
2. Right click on the SendKeys command.
3. Select Start Recording.
4. On the Samsung A707, press Open menu.
5. Select Messaging.
6. Right click on the SendKeys command.
7. Select Stop Recording.
8. Click Save and Close.

5.9.4 Verify Messaging

Now verify the Messaging screen on the device by using a WaitImage command.

1. Drag another WaitImage command into the editor.
2. Highlight part of the screen that verifies that the messaging screen is open.
3. Change the label of the WaitImage command, enter Verify Message in the Label field.
4. Click Save and Close.

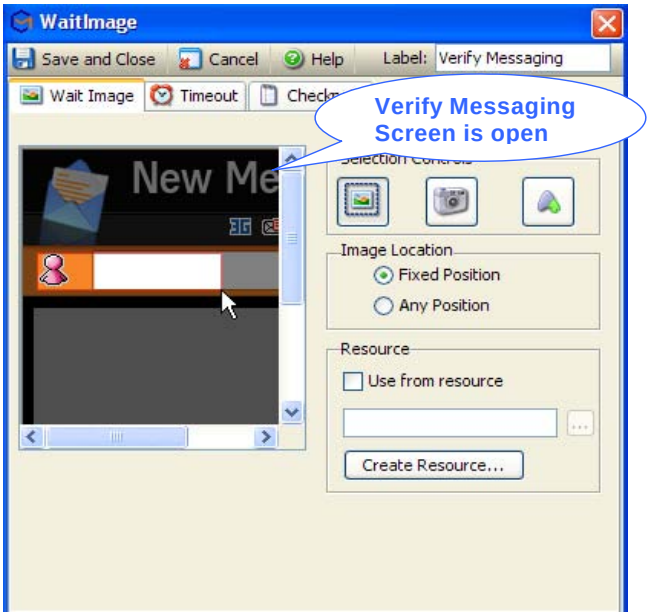


Figure 5-35 Verify Messaging Screen

In this example, the messaging screen is stored in the WaitImage command. It is important to use WaitImage commands along the way to ensure that the script is executing properly and also to make sure that commands are not executed on the device before it is ready to accept input.

5.9.5 Select New Message

Navigate to the New Message Screen:

1. Drag a SendKeys command to the green target.
2. Right click on the SendKeys command.
3. Select Start Recording.
4. On the Samsung A707, select New Message from the menu.
5. Right click on the SendKeys command.
6. Select Stop Recording.
7. Change the label of the SendKeys command by entering Select New Message in the Label field.
8. Click Save and Close.

5.9.6 Insert Phone Number

Navigate to the New Message Screen:

1. Drag a SendKeys command to the green target.
2. Right click on the SendKeys command.
3. Change the label of the SendKeys command by entering Insert Phone Number in the Label field.
4. Click Save and Close.

You'll return to this SendKeys to select a Macro parameter after the Num and Msg variables are defined for the Send SMS Macro resource in Section 5.9.12.

5.9.7 Insert TXT Message

Enter the Text Message into the Message Field:

1. Drag a SendKeys command to the green target.
2. Right click on the SendKeys command.
3. Change the label of the SendKeys command by entering Insert TXT Message in the Label field.
4. Click Save and Close.

You'll return to this SendKeys to select a Macro parameter after the Num and Msg variables are defined for the Send SMS Macro resource in Section 5.9.12.

5.9.8 Send TXT Message

Send the Text Message to Weather.com:

1. Drag a SendKeys command to the green target.
2. Right click on the SendKeys command.
3. Select Start Recording.
4. On the Samsung A707, navigate to the Send button and select it.
5. Right click on the SendKeys command.

6. Select Stop Recording.
7. Change the label of the SendKeys command by entering Send TXT Message in the Label field.
8. Click Save and Close.

5.9.9 Verify Message Sent

Now as soon as the message is sent, you're ready to capture a reference image that is compared to the Messaging Inbox.

1. Add a WaitImage command to the script editor.
2. Double click the WaitImage command to view the Properties.
3. Use your mouse and select an area of the message alert screen to create your reference image.
4. Change the label of the WaitImage command by entering Verify Message Sent in the label field.
5. Click Save and Close.

Figure 5-36 shows the Message Inbox screen, indicating a message was sent successfully.

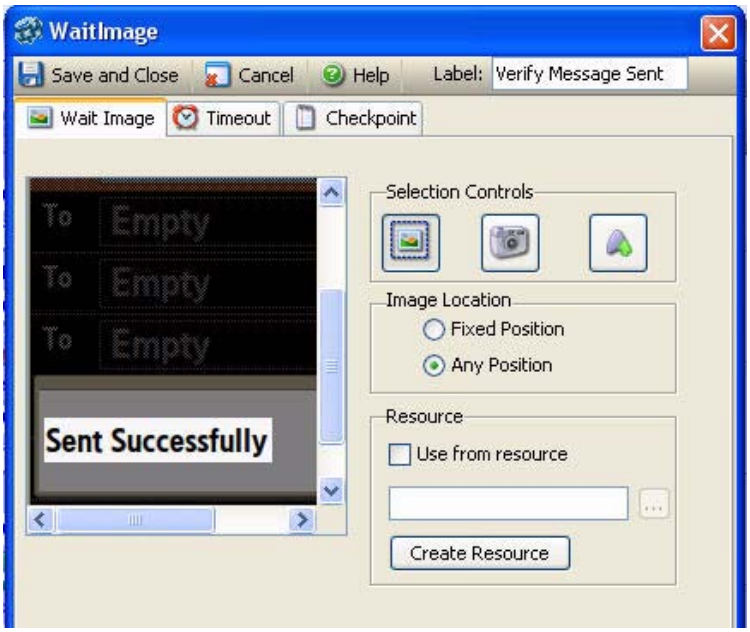


Figure 5-36 Message Sent Screen

5.9.10 Return to Idle

Return to Idle follows the same steps needed to return the device to the idle screen. This procedure should be easy for you by now.

1. Add a SendKeys command to the script editor.
2. Right click on the SendKeys command and select Start Recording.
3. Now press the Power key many times on the Samsung A707.
4. Right click on the SendKeys command and select Stop Recording.
5. Change the label of the SendKeys command by entering Return to Idle in the Label field.
6. Click Save and Close.

The device has now returned to the idle screen. It is important to return the device to an idle state where your next test script can start from. As a general rule, add the steps at both the beginning and the end of the script to return the device to a known state.

5.9.11 Success

Drag the Success command into the script. This command sends out a Status message on script execution. You can enter the message that you want to receive when the script runs to completion successfully.

5.9.12 Set Up Macro Parameters in Send SMS

The Send SMS Macro resource is designed generically to support SMS transactions. Two local variables, Num and Msg, are created in the Macro resource.

Figure 5-37 shows the Script Properties screen for the Send SMS Macro.

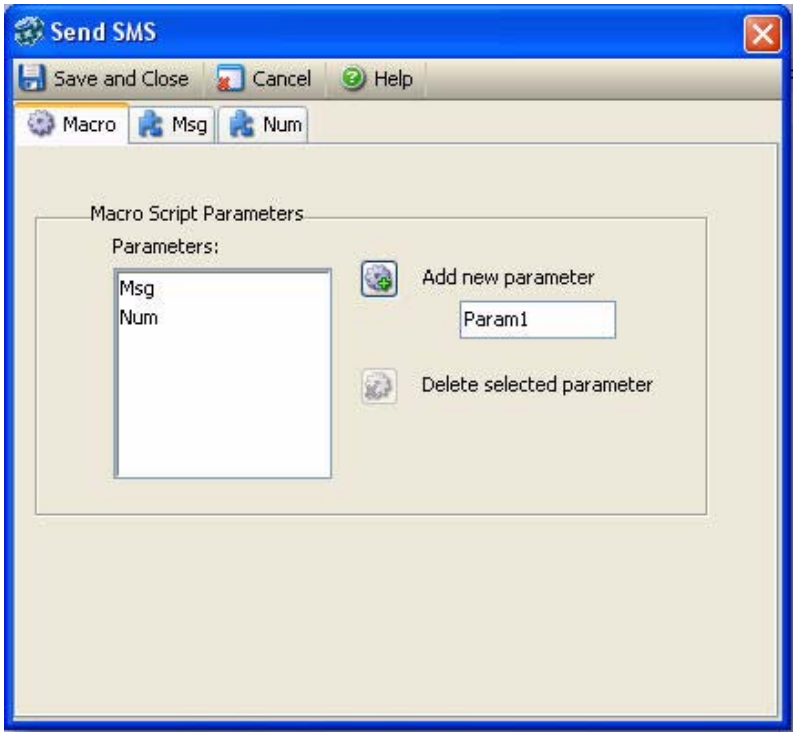


Figure 5-37 Send SMS Local Variables

To create local variables follow the procedures below.

How to Define the Variables:

1. Right click the Send SMS Macro in the Explorer. The drop-down menu is displayed.
2. Select Edit Macro Parameters.
3. Enter the name of the variable. In this case, enter Num.
4. Click Add new parameter. A Num tab is added to the Properties screen.
5. Select the Num tab.
6. Select the variable type. In this case, select Numeric value.
7. Enter the shortcode 44782 for a weather forecast at weather.com.
8. Click Save and Close.

Now create the Msg variable by following the same steps outlined above.

1. Right click the Send SMS Macro in the Explorer. The drop-down menu is displayed.
2. Select Edit Macro Parameters.
3. Enter the name of the variable. In this case, enter Msg.
4. Click Add new parameter. A Msg tab is added to the Properties screen.
5. Select the Msg tab.
6. Select the variable type. In this case, select Numberic value.
7. Enter the keyword. In this case, enter a zip code for the weather forecast request.
8. Click Save and Close.

5.9.13 Insert Local Variables in the SendKeys

Now that the local variables are defined for the Send SMS Macro, follow the procedure to insert the variables into the SendKeys called Insert Phone Number and Insert TXT Message.

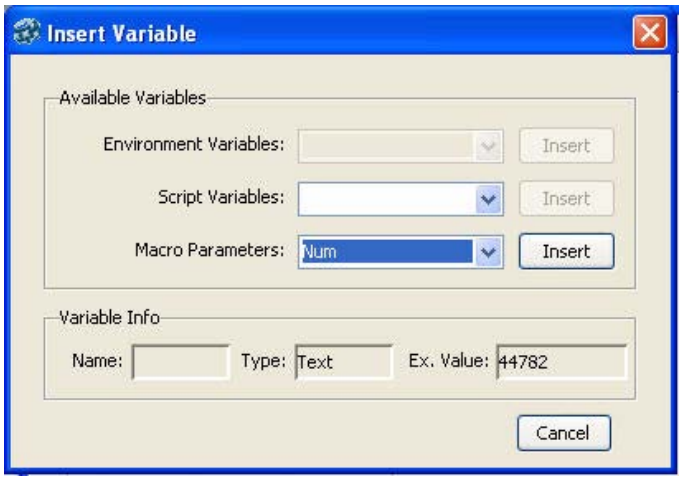


Figure 5-38 Insert Num Variable in SendKeys

How to Insert a Variable in a SendKeys:

1. Right click the SendKeys command. In this case, Insert Phone Number.
2. Click the Puzzle piece next to the Text To Send field. The Insert Variable dialog appears.
3. Click the drop-down menu to insert the correct variable for the SendKeys. For example, select Num for the Insert Phone Number SendKeys.
4. Click Insert. The variable Num is inserted into the Text To Send field.

The Num variable appears in the following format: %Num%.

Save your work by right clicking on the Macro instance and selecting Save changes.

5.9.14 Associate Macro Resources in the Test Procedure

To associate Macro resources in the high-level script, you must return to the Automation view and select the Macro resource through the Properties screen of the Macro call.

How to Associate a Macro Resource to a Macro Call:

1. Ensure that you are in the Automation view.
2. Select the script called Send SMS in the Explorer. The Test Procedure is displayed in the workspace.
3. Double click the Send SMS Macro call. The Properties screen is displayed.
4. Next to the Macro Selection window, select the Building block icon. This action displays the Resource Libraries in the Resource Manager.
5. Navigate the libraries to find the Send SMS Macro resource.
6. Highlight Send SMS in the window.
7. Click Select. This action associates the Macro resource with the Send SMS Macro call.

When the Macro resource is selected, the two Macro parameters that are defined for Send SMS, Num and Msg, are displayed in the Macro Parameters drop-down list on the Properties screen.

Figure 5-39 shows the Send SMS Macro resource associated to the Send SMS Macro call. The Macro Parameter Msg is also showing on the screen.

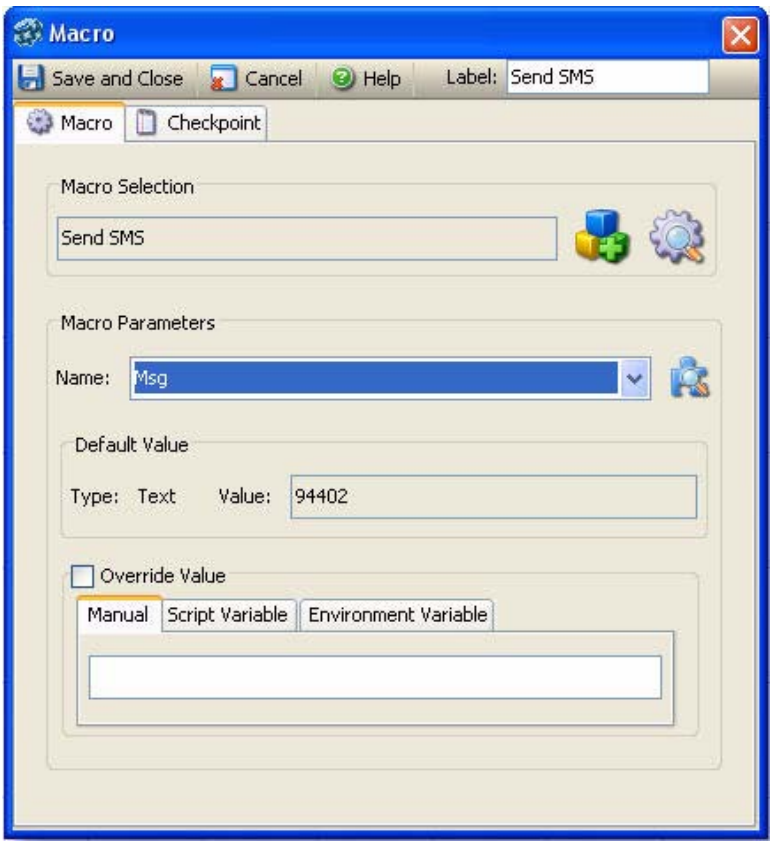


Figure 5-39 Properties Screen for the Send SMS Macro Call

When the Send SMS script is executed, the values for the Num and Msg variables are evaluated and implemented to send a keyword to the weather.com shortcode 44782.

5.9.15 Save Changes to Send Message Script

Now save your changes to the Send Message script by right clicking on the script and selecting Save Changes in the drop-down menu.

DeviceAnywhere Studio has version control built into the product. Scripts are automatically saved to the database as you are working. However, Saving Changes writes an official copy of the script to the database for the script History. When you write the script to the database, the Lock icon is removed from the script to indicate that it is released for editing. Any team member can access the script at this point.

For scripts, right click the script in the Explorer and select the Lock/Edit command to edit the script further after saving changes.

5.9.16 Play Your New Send Message Script

Play your script by pressing the Play button in the toolbar. Make sure your script executes from start to finish. If it doesn't complete successfully, take the time to understand why the script fails and make the appropriate modifications.

Key points:

- The WaitImage command checks to see if the defined reference image appears on the device.
- The reference image can be looked for at a fixed location or any position. Any position is useful when looking for images that may move around the screen (dynamic lists for example).
- The WaitImage command continues to look for the image until the Timeout value is reached.
- It is important to use WaitImage commands along the way to ensure that the script is behaving properly and also to make sure that commands are not executing on the device before it is ready to accept input.
- You don't have to run the script from the beginning. You can run from any intermediate step in the script. This is great for debugging and entering copied text into a device.
- Key Mapping is an important feature within *DeviceAnywhere*. When entering any text into the device as part of a script, you must select the correct mode (Alpha for text, Numeric for numbers, and Web for URL entry). The default Key Mode is Alpha.
- It is important to return the device to a state where your next test script will start from. As a general rule, add the steps at the beginning of the script to return the device to a know state and also do it at the end of the script.

5.10 Work with Variables in a Script

DeviceAnywhere Pro enables you to define local and global variables in the design of test scripts. The variables in the *DeviceAnywhere Pro* scripting language provide a level of abstraction such that a value can be passed to the script at run-time. This feature facilitates the reuse of scripts with different values passing to the script. In addition, it supports the ability to drive a script with a dataset.

The *DeviceAnywhere Pro* scripting language supports three types of variables:

- True/False boolean expressions
- Numeric values
- Text strings

The procedure below describes the workflow to create a variable and implement it in a test script.

1. Define the variable. This action consists of giving the variable a name and defining its type.
2. Define the value that is passed to the variable at run-time. The value will correspond to a true or false boolean expression, a numeric value, or a text string.
3. Associate the variable to a step in the script. For example, use the Properties screen of a SendKeys command to associate a variable to the script. At this step in the script, the variable is evaluated and used in script execution.

5.10.1 Variable Features

Three features of the scripting language enable you to create variables:

- SendKeys, WaitImage, and WaitEvent script commands support the use of local variables through the DataSet tab in the Properties screen.
- Macro parameters can be defined for a Macro resource through the Macro Properties screen.
- The Variables icon in the Automation view is used to create global variables. These variables can be associated to any command in the Test Procedure. They are also called Environmental variables.
- The SetVariable command enables you to define static variables within a script. SetVariables are used most often to define Boolean case logic for branching expressions.

5.10.2 DataSet Tab in Script Properties

Figure 5-40 illustrates the DataSet tab in Script Properties. Two variables, Phone Num and Message, have been defined, and several rows have been added to the datase. Each row contains a record—a phone number (numeric value) and a message (numeric or text value).

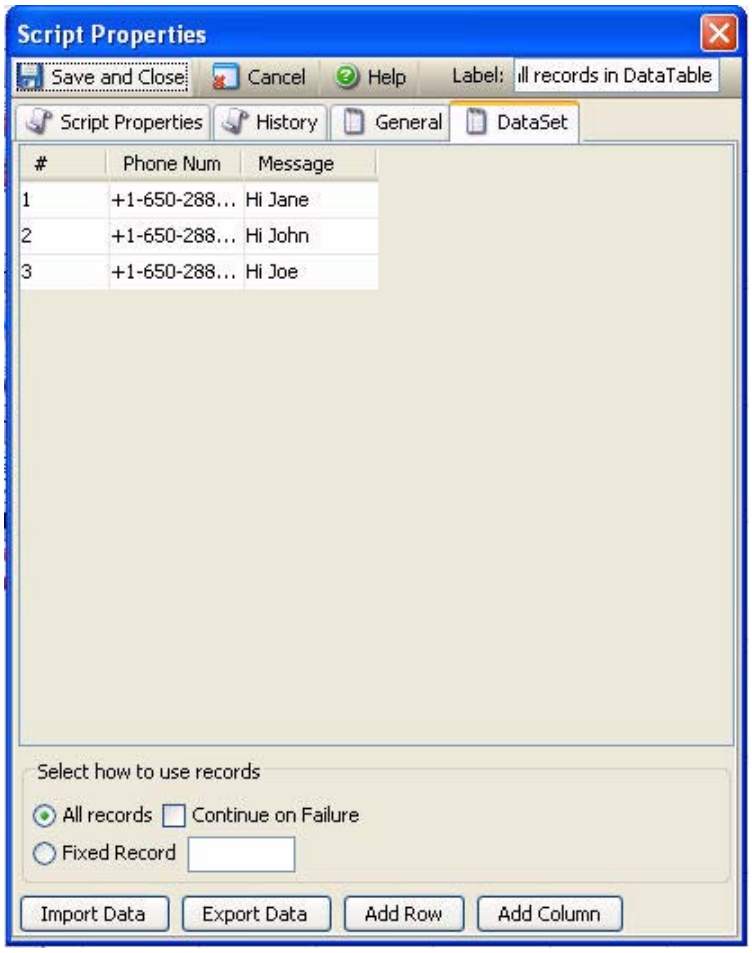


Figure 5-40 Script Properties—Dataset Tab

The DataSet tab supports the following options:

Import Data—This option enables you to import an .csv file that is used as the dataset.

Export Data—This option enables you to export the dataset that you created in the DataSet tab to an excel file on your computer.

Add Row/ Add Column—These options enable you to create the values that will be evaluated for each variable at run-time and define the variable names in the dataset.

Select how to use records—These options determine the mode of execution of the script. All records means all the records in the dataset are executed. The script automatically loops through the dataset, executing the values in each record.

Continue on Failure allows the script to continue executing, even though a failure occurred. Fixed Record enables you to specify a fixed number of records to be used during the execution of the script.

5.10.3 Create Variables for a Dataset

The description below provides the steps needed to create a small dataset as shown in Figure 5-40. As an example, two variables Phone Num and Message are defined for a Send Message script.

To define variable names, follow the steps below:

1. In the Automation view, right click a script in the Explorer. The Script menu is displayed.
2. Select Properties.
3. Select the DataSet tab.
4. Click Add Column. The row header dialog box appears.
5. Enter the name of the row header. This name becomes the variable name that you use in the script. In this case, enter Phone Num.
6. Click OK.
7. Click Add Column.
8. Enter the name of the row header. In this case, enter Message.
9. Click OK. Two variables are now defined.

To insert values into the dataset, follow the steps below:

1. Click Add row. A row appears under the column headers.
2. Click the cell under Phone Num.
3. Enter the phone number where a message is to be sent.
4. Click the cell under Message.
5. Enter the message to be sent.
6. Continue adding values in the Phone Num and Message columns to complete the dataset.
7. Click Save and Close to save the dataset.

5.10.4 Implementing Variables in a Script

In Section 5.10.3 you created two variables in the Script Properties screen and entered phone numbers and messages to complete a dataset. In this section, you implement the variables in a Send Message script. You need to open a completed script in the workspace before implementing variables.

Figure 5-41 illustrates a Send Message script with a sequence of SendKeys and WaitImage commands.

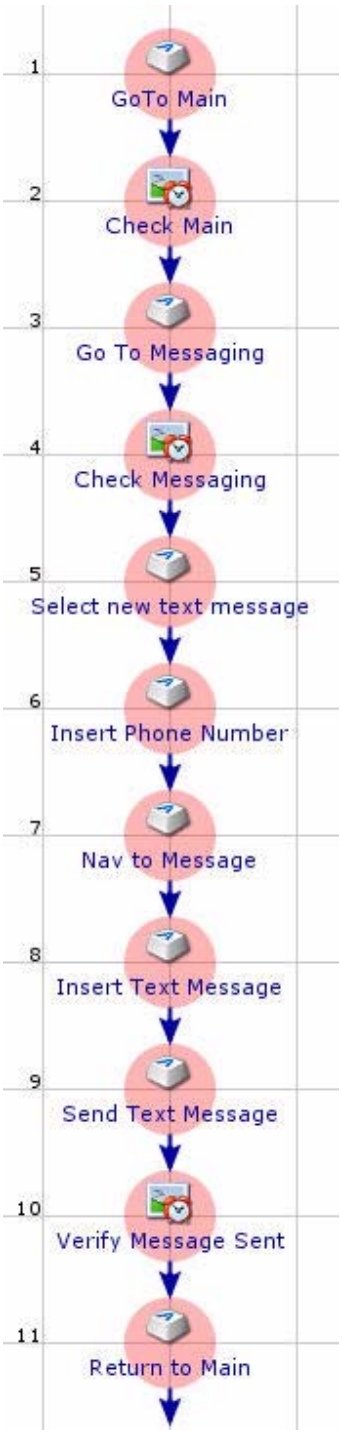


Figure 5-41 Send Message Script with Variables

The SendKeys command supports the insertion of a variable in the Text To Send field. In the Send Message script, two SendKeys will use the variables Phone Num and Message:

- Insert Phone Number—Step 6 uses the Phone Num variable.
- Insert Text Message—Step 8 uses the Message variable.

To implement a variable in a script, follow the steps below:

1. In the Automation view, select a script. The script is displayed in the workspace.
2. Right click the SendKeys command where the variable is used. In this case, right click Insert Phone Number.
3. Select Properties.
4. Near the Text To Send field, click on the Puzzle piece. The Insert Variable dialog box is displayed.
5. For script variables, click on the drop-down list. All the variables created in the Script are displayed.
6. Select Phone Num.
7. Click Insert.

The variable name Phone Num is inserted into the Text To Send field. Now repeat this procedure to implement the Message variable in the SendKeys Insert Text Message.

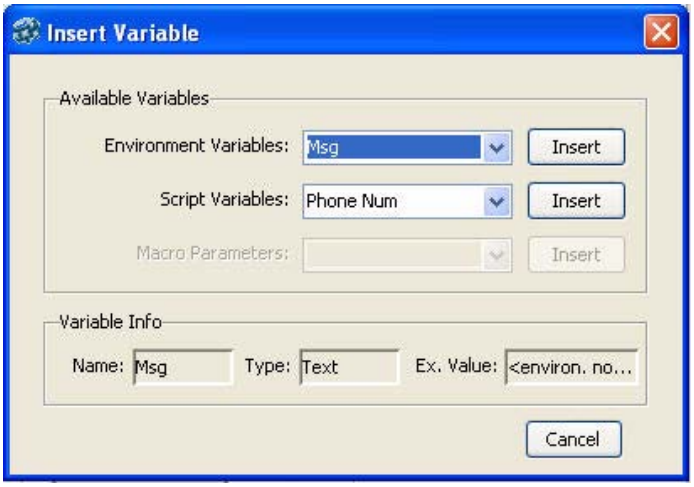


Figure 5-42 Script Variable—Phone Num

When the Send Message script is executed, a new record is passed to the script and stored in the variables Phone Num and Message.

Right click the script in the Explorer and select Save Changes. This action saves the variables in your script.

5.10.5 Check How to Use Records

The script evaluates the variables in the dataset based on the settings in How to use records on the DataSet tab.

To check the current setting, follow these steps:

1. Right click the script in the Explorer.
2. Select Properties. The Script Properties screen is displayed.
3. Select the DataSet tab.
4. Select an option in the How to use records window. All records is set by default.
5. Click Save and Close.

5.10.6 Create Macro Parameters

In this section, you create and implement the variables in a Macro. Variables are also known as Macro parameters in the Resource Manager.

Figure 5-43 shows the Send SMS Macro in the Resource Manager view. Send SMS is a Macro that sends a keyword to a service provider shortcode. In this case, two variables, Num and Msg, are created to store the keyword and shortcode.

In the Send SMS Macro, two SendKeys will use the variables Num and Msg:

- Insert Phone Number—Step 6 uses the Num variable.
- Insert TXT Message—Step 8 uses the Msg variable.

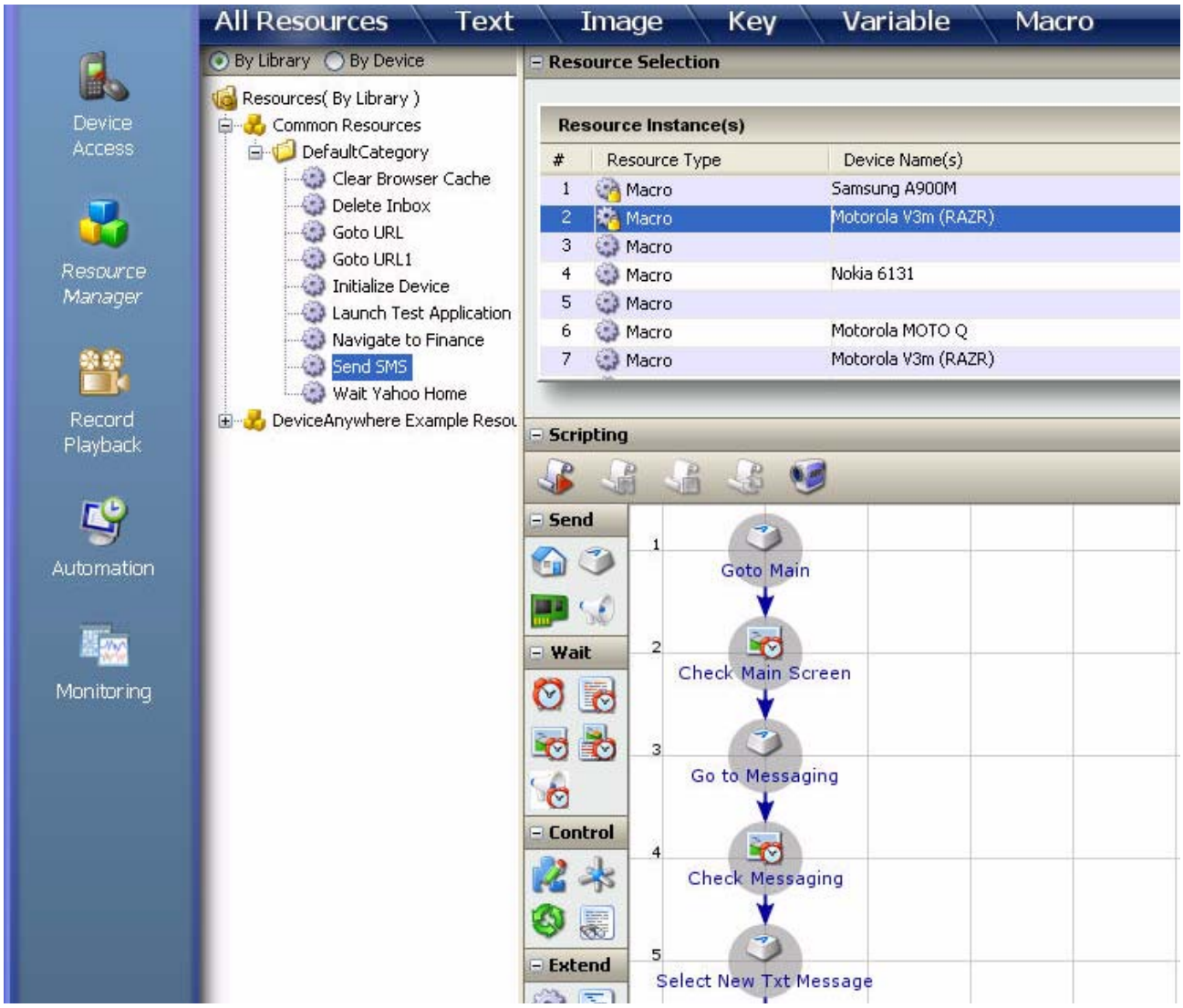


Figure 5-43 Send SMS Macro in the Resource Manager View

5.10.7 Create Variables in a Macro Resource

To create and define variables in a Macro resource, open the Edit Macro Parameters dialog box.

Right click the Macro in the Explorer and select the Edit Macro Parameters option.

Figure 5-44 shows the Edit Macro Parameters screen. Msg and Num are two variables defined for the Send SMS Macro.

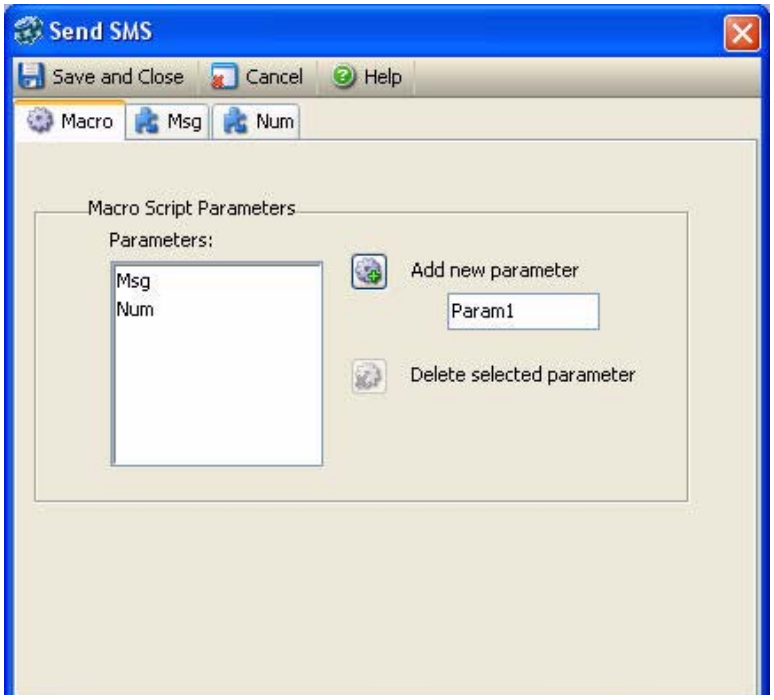


Figure 5-44 Edit Macro Parameters

To create a parameter (e.g. variable) in the Macro, follow the procedure below:

1. In the Edit Macro Parameter screen, enter the name of the parameter in the text box. In this case, enter the name Num.
2. Click Add new parameter. A tab for the parameter Num is displayed.
3. Click on the Num tab. This screen enables you to define the type of variable. Choose a true/false boolean expression, numeric value, or text string.
4. Select Numeric value.
5. Enter a numeric value in the entry field. In this case, enter 44782, which is a short-code to request a weather forecast.
6. Click Save and Close.

Now open the Insert TXT Message SendKeys command. Repeat the procedure above to create the Msg variable. This time enter the keyword 94402. This keyword requests for the weather forecast of the given zip code.

Figure 5-45 shows the Num tab in the Send SMS Macro.

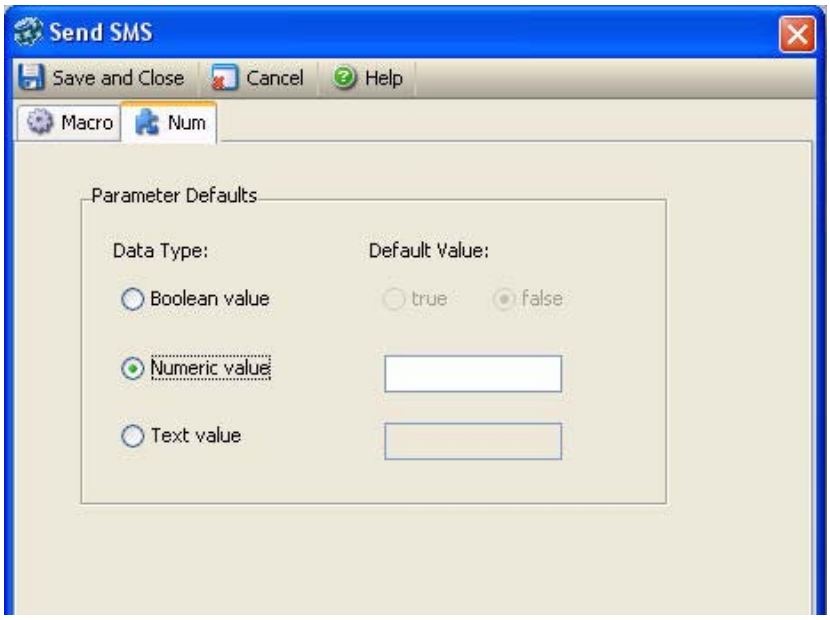


Figure 5-45 Parameter (Variable) Type

5.10.8 Create Global Variables

Global variables can be associated to the Macro call in the Test Procedure or to a SenKeys in a Macro resource.

5.10.9 Set Up a Global Variable

How to Create a Global Variable:

1. Right click on the Variables icon in the Explorer.
2. Select New Variable.
3. Enter the name of the new variable. In this case, enter Phone Num.
4. Create a second global variable called Msg by repeating Step 1, Step 2, and Step 3.

5.10.10 Implement Global Variables in a Test Script

You created two global variables in the previous section, now define their values.

1. Right click Phone Num in the Explorer.
2. Set the type of variable: Boolean True/False, Numeric, or Text. In this case, choose Numeric value.
3. Enter the value: 44782.
4. Select Save and Close.

Repeat this procedure and select the Msg global variable in the Explorer. This time specify a text string and enter the keyword (zip code) for Send SMS.

5.10.11 Associate a Global Variable

Global variables are associated to a Macro call or any SendKeys command in the high-level or low-level script. This feature is accomplished through the

Figure 5-46 illustrates the Override Value checkbox. When Override Value is checked, the variable is evaluated based on the Manual, Script Variable, or Environment Variable selected.

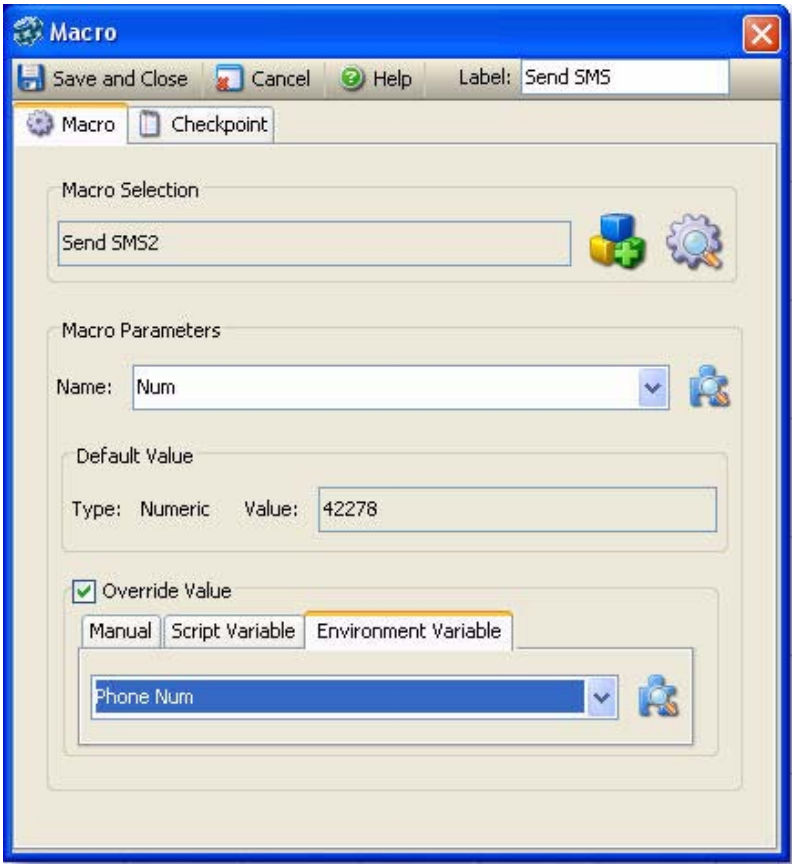


Figure 5-46 Variable Override

5.11 Test Suites

This section shows you how to take advantage of the scheduling features in *DeviceAnywhere Pro*, which enable you to run tests offline (overnight or while you are at lunch, and so on). Test Suites provides the ability to logically organize multiple test scripts into a Test Suite, to schedule tests to run at specific times, and to view Test Suite results.

The general procedure for test automation is to create individual test scripts and then group those test scripts for batch execution. In previous sections, you became familiar with scripting. Now it's time to take your individual test scripts and group them into a Test Suite.

This section describes the procedures to do the following:

- Create a Test Suite
- Run a Test Suite
- Schedule a Test Suite to run automatically at specified times
- View the results of the Test Suite

5.11.1 Test Suite View

Figure 5-47 illustrates the Test Suite view. In this example, a Test Suite called MOT L6 Test is shown in the Explorer.

Figure 5-47 Test Suite View

The Test Suite View has three windows to manage Test Suites:

- The Add, Delete, Sequence window shows you the test scripts that have been added to the Test Suite. The buttons on the bottom of the window allow you to add, delete, and sequence test scripts. Use the Up and Down buttons to sequence the test scripts in the list.
- The Configure window (not shown here) enables you to control test script execution based on the return status of the test script. When an individual test script executes, it returns with a Success, Failure, Exception or Interrupted status. The Configure window enables you to specify a final action based on each status. For example, when a test script returns a Failure status, you can select abort or continue from the drop-down menu in Final Action column. In this way, the Test Suite aborts or continues immediately after the Failure status is returned.
- In addition, you can add a value in the Times field to set the number of iterations of the test based on status. For example, when you want to run the Lesson 5a test script in the Test Suite a second time because it fails, increase the number in the Times field to 1.
- The Current Script window (not shown here) enables you to view the current script that is executing.

5.11.2 Creating a Test Suite

Figure 5-48 illustrates the New Test Suite command in the drop-down menu.

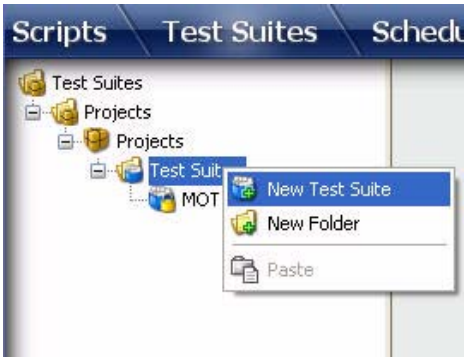


Figure 5-48 Create a New Test Suite

How to Create a New Test Suite:

1. Right click on the Test Suites tab in the Automation view.
2. Select New Test Suite.
3. Enter the name of the new Test Suite. In this case, MOT L6 Test.
4. Right click on a device in the Device list.
5. Select Add to Test Suite.
6. Right click on the device in the workspace. The Device menu appears.
7. Select Access Device.
8. Select Acquire Device.
9. Select Save Changes.

After creating a new Test Suite, the Test Suite view is empty. The device you added to the Test Suite is shown in the far right panel. Now you are ready to select one or more scripts to include in the Test Suite.

5.11.3 Adding Scripts to a Test Suite

The Add, delete, sequence window enables you to select the scripts to run in a Test Suite.

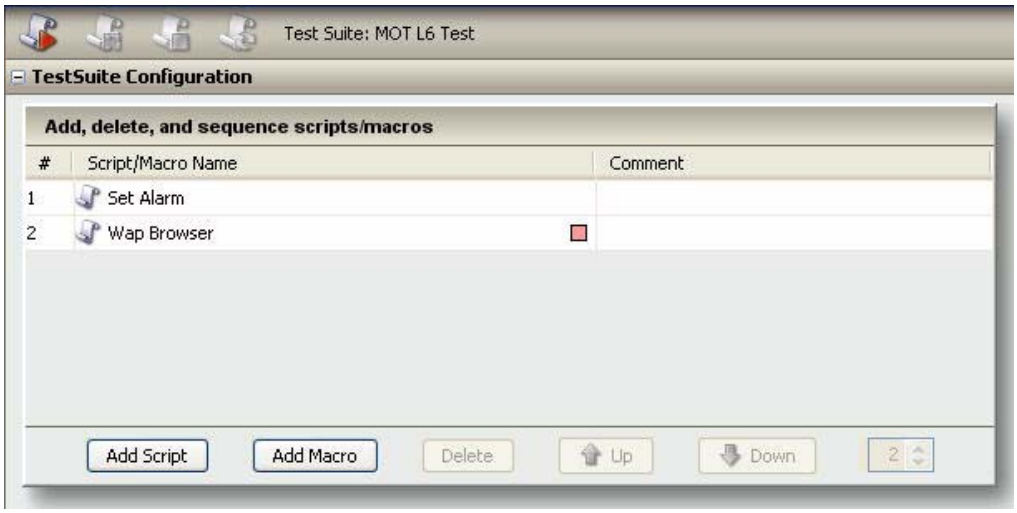


Figure 5-49 TestSuite Configuration

How to Add a Script and/or Macro:

1. Click Add Script. The Select Scripts browser appears.
2. Navigate through the script folders and expand each level to find the script you want to add to the Test Suite.
3. Highlight the script with the cursor.
4. Click Add. The script is added to the Test Suite.

You can add a Macro to a Test Suite just like regular scripts.

Use the buttons on the bottom of the window to sequence the scripts and Macros. First, highlight the script with the cursor, and then click the Up or Down buttons to change the sequence.

When you want to delete a script from the Test Suite, highlight the script and click the Delete button.

5.11.4 Configuring a Script or Macro

In the Configure window, you can program the exit path for scripts and/or Macros based on four error conditions.

In the Return code column, four error conditions are shown.

The error conditions are described as follows:

- Exception—This error condition indicates there is a problem with the test script. The test script fails to compile.
- Failure—This error condition indicates the script did not run to completion. This failure signals a network or mobile application error occurred.
- Success—This error condition indicates the script ran to a successful completion.
- Interrupted—This error condition indicates the execution was interrupted in the middle of the script. This is generally a user-initiated interruption.

Figure 5-50 shows the Configure Script/Macro window. The Exception error condition is programmed to launch the Initialize Device Macro when the error is encountered.

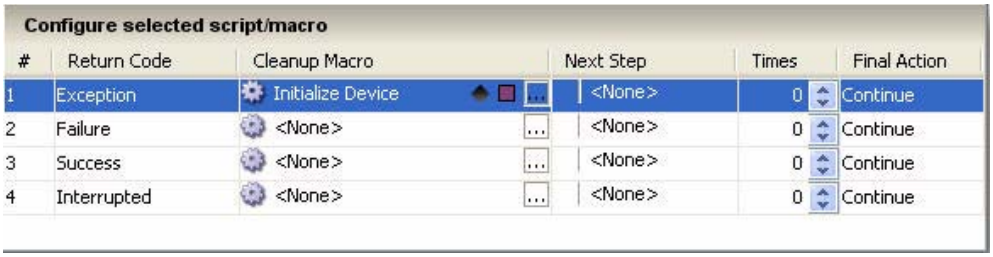


Figure 5-50 Configure Script/Macro

In the Cleanup Macro column, you have the option to select a Macro that will run when the corresponding error condition is encountered. Press on the search button to open the Select or Clear Macro dialog box.

How to Select a Cleanup Macro:

1. Click on the search button (dotted line).
2. Navigate and expand the folders in the Resource Manager.
3. Highlight the Macro you want to initiate when the error condition occurs.
4. Highlight the device in the Slot position. In this case, Slot 1.
5. Click Select. The Cleanup Macro is set.

Repeat this procedure to set a Cleanup Macro for each error condition in the list.

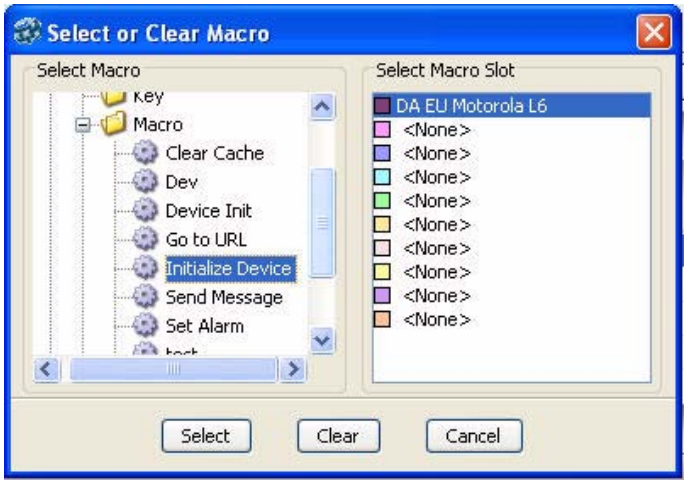


Figure 5-51 Select or Clear Macro Dialog Box

In the Next Step column, you have the option to select a script or Macro that defines the next step in the execution path for the Test Suite. Place your cursor in the field to display a drop-down menu of the scripts and Macros. Select a script or Macro from the menu.

The Times column enables you to set a value to reiterate the script when an Exception or any error condition occurs.

In the Final Action column, you can program the Test Suite to move on to the next script or abort the Test Suite.

5.12 Playing a Test Suite

In the Current Script/Macro window, you can watch a Test Suite as it executes. Under Script Execution, look for the name of the active script and the test step to follow along.

In order to play the Test Suite, like scripts and macros, you must define and acquire a device.

When a device is acquired, the Play button in the toolbar changes to green, indicating the Test Suite is ready for execution.

Click Play to execute the Test Suite. The Stop icon in the toolbar then becomes active, indicating the Test Suite is running.

Figure 5-52 shows the current script that is executing.

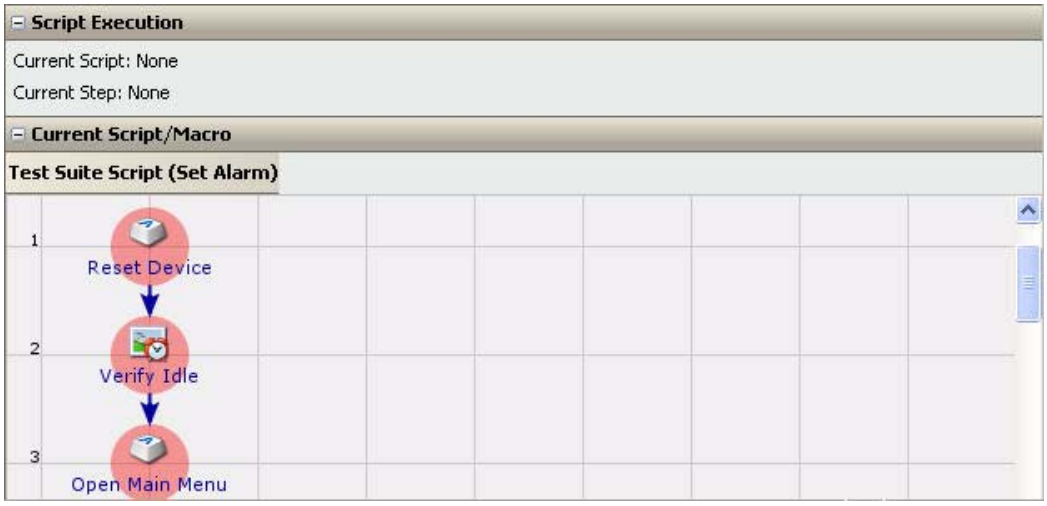


Figure 5-52 Current Script/Macro View

5.12.1 Viewing Test Suite Results

When the Test Suite finishes execution, a dialog box appears that shows a summary of the Test Suite results. To view a detailed report, click the checkbox before the Upload results as field, enter a name for your test results, and click the Upload button.

Once the upload is complete, click the View Now button to be taken to the *MyDeviceAnywhere* website to view the Test Suite results.

Within the Test Results tab of the *MyDeviceAnywhere* website, you see a high-level report of your test results. Click the View Details link to view a more detailed report of the Test Suite results.

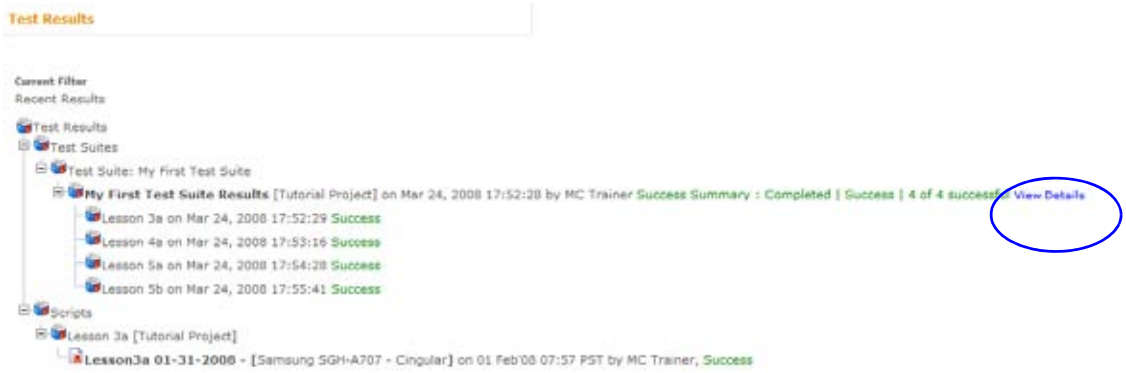

Test Results
Current Filter
Recent Results
Test Results
Test Suites
Test Suite: My First Test Suite
My First Test Suite Results [Tutorial Project] on Mar 24, 2008 17:52:28 by MC Trainer Success Summary : Completed | Success | 4 of 4 successful View Details
Lesson 3a on Mar 24, 2008 17:52:29 Success
Lesson 4a on Mar 24, 2008 17:53:16 Success
Lesson 5a on Mar 24, 2008 17:54:28 Success
Lesson 5b on Mar 24, 2008 17:55:41 Success
Scripts
Lesson 3a [Tutorial Project]
Lesson3a 01-31-2008 - [Samsung SGH-A707 - Cingular] on 01 Feb'08 07:57 PST by MC Trainer, Success

Figure 5-53 High-level Report of Test Suite Results

The result details view (not shown) includes the following information:

- Script Info—This displays the name of the script, the device make and model, and the carrier.
- Run Info—This displays the summary of the result; project name, test environment, outcome, run by, start time, duration and description.
- Step-By-Step—This section provides step-by-step results of the entire script. It lists Data, Start, End Time and Images against each step to give you a clear picture of what happened during the test. Clicking on View will show the resulting images for the step.
- Follow-Up History—Comments posted by different viewers on the test are available here.
- Follow-Up Comments—You can add your comments to the various runs here, and save them by clicking the Save button on the page.

5.12.2 Sharing Results

The following options are available for sharing results:

- Email Results (secure)—This option enables you to send your results to other *MyDeviceAnywhere* users. Users receive a link to the Results page, which will only be accessible when the users have an account with *MyDeviceAnywhere*.
- Email Results (standard)—This option enables you to send your results to anyone (*MyDeviceAnywhere* account holders and others).
- Save—Pressing this icon saves any comments entered.

Sharing results enables you to choose any one of the available formats, such as XML file, HTML (with images) or HTML (without image). You can click on the button Export to export the results.

5.13 Scheduler

The Schedule tab enables you to open a calendar and schedule Test Suites to run automatically. It supports a calendar and Schedule Test screen to set parameters for automated testing.

Figure 5-54 illustrates the Schedule view.

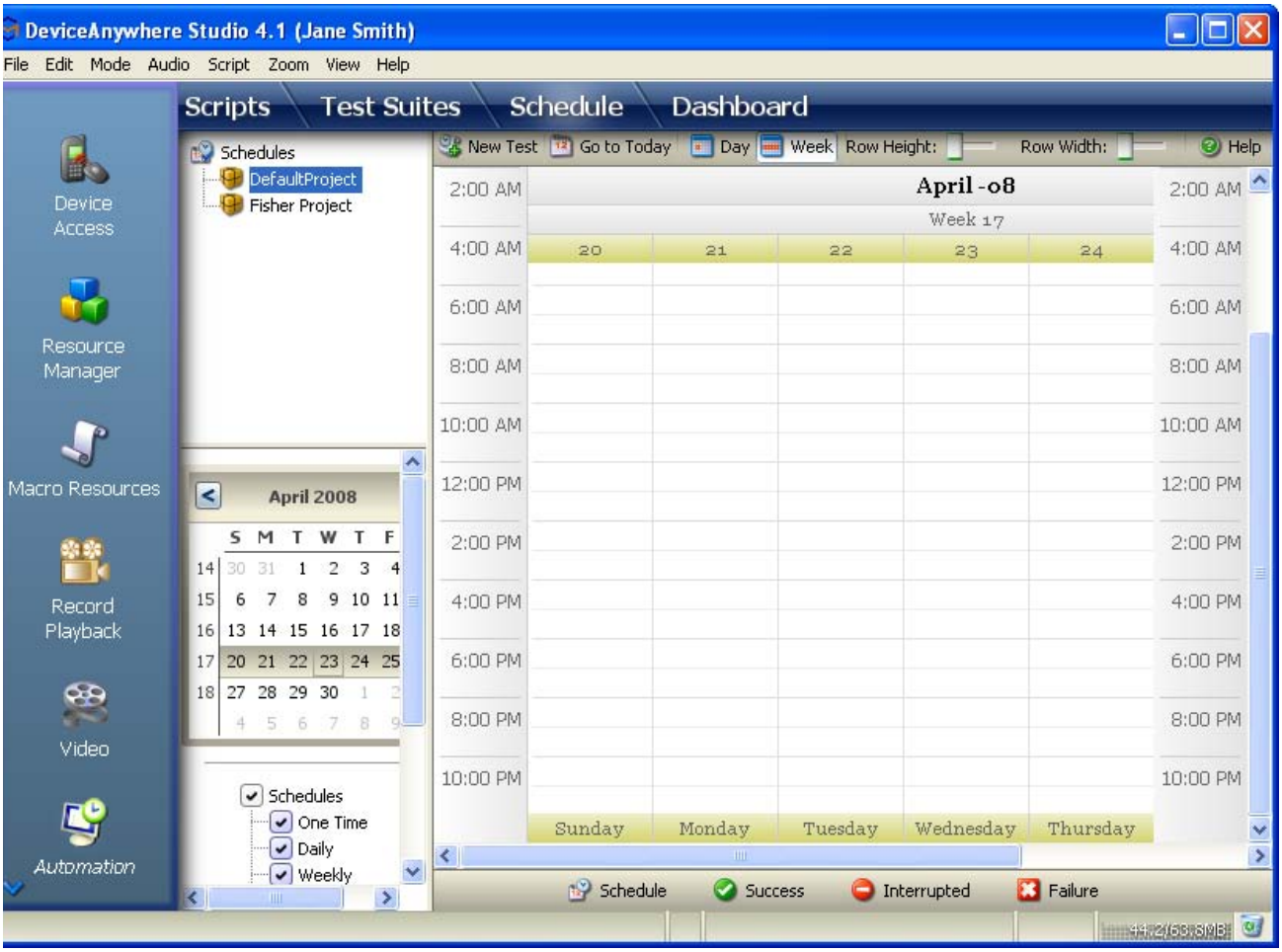


Figure 5-54 Schedule View

Schedule Editor shows a grid the date and time. This grid contains cells with scheduled entries for the project you have select.

5.13.1 Schedule Editor Toolbar

The following commands are available in the Schedule Editor Toolbar:

- New Test—Creates a new scheduled entry.
- Go to Today— Gives the status of today's tests.
- Day—Changes the grid view to accommodate only one day.
- Week—Changes the grid view to accommodate a week.
- Row Height—You can adjust the height of the workspace rows.
- Row Width— You can adjust the width of the workspace rows.
- Help—Launches the *DeviceAnywhere Studio* online help.

5.13.2 Scheduling Test Suites

Figure 5-55 shows the Schedule Test Suite screen.

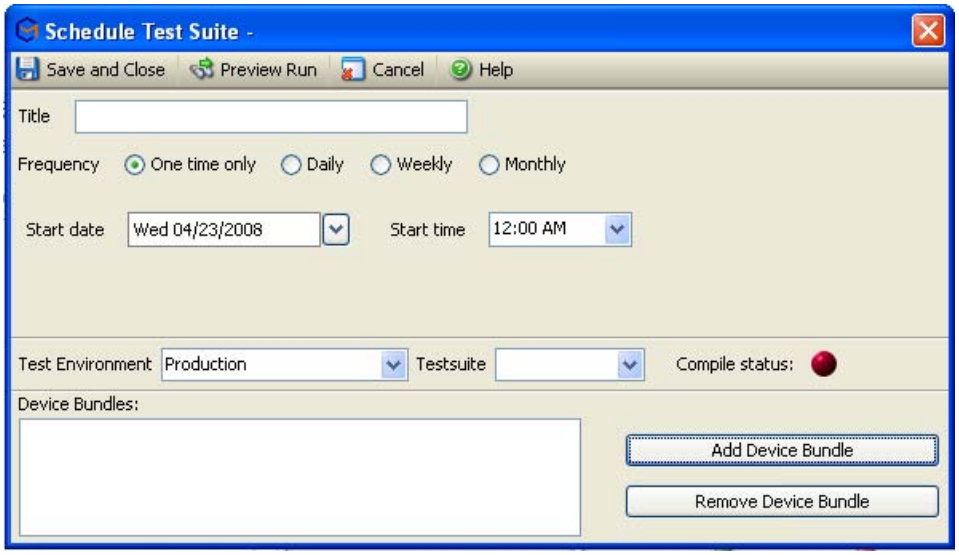


Figure 5-55 Schedule Test Suite Screen

To open the Scheduler Editor and select parameters, do the following:

1. Click on the Schedule tab to go to the Schedule Editor.
2. Right click on Projects in the Explorer.
3. Select New Test.
4. In the Schedule Test Suite dialog box, enter a title for your test in the Title field.
5. Specify the frequency that it will execute.
6. Select a Start Time.
7. Select a Test Suite by name from the Test Suite drop-down menu.
8. Click the Add Device Bundle button.
9. Select the first line for a device you are working with.
10. Select Save and Close to schedule the test.

A description of the parameters used to schedule an automated test are given below:

- Title—The title for the test.
- Frequency—To select if the test would run once or periodic.
- Start date—The date to begin the test.
- Start time—The time to begin the test.
- Test Suite—The test suite to be run in the scheduled test.
- Device Bundles—The test that is run on the devices that are present in the Device Bundles window.
- Test Environment—The environment in which testing is done.

Enter the title of the test and select the frequency: One Time Only, Daily, Weekly or Monthly. Next, you need to set the date and time (by default the current date is selected). Then select the test suite you will be executing (all the available test suites are listed in the list box to the right).

5.13.3 Create a Device Bundle

When you select a Test Suite from the drop-down menu, you already have a number of devices added to the test scripts. However, you must specify the devices explicitly to be used in the Test Suite.

This procedure uses the Script Slot Configuration screen to assign devices to a slot and then add the devices to the Test Suite. Figure 5-56 shows a Script Slot Configuration screen when you begin to specify the settings of a Test Suite.

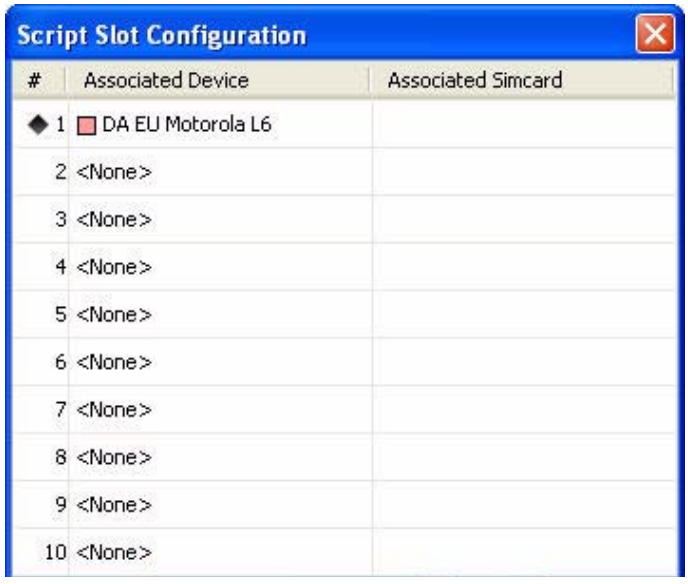


Figure 5-56 Create a Device Bundle

How to Create a Device Bundle:

1. To add a device to the Device Bundle, click the Add [to] Device Bundle button. The Script Slot Configuration screen appears.
2. Place your cursor in the first line of the screen under Associated Device. This action opens a drop-down menu. All the devices added to the test scripts are represented.
3. Select a device for the first slot.
4. Repeat Step 2 and Step 3 for all devices that you'd like to add to the Device Bundle.
5. Click Close.

All the devices selected for the Test Suite are now listed in the Device Bundles window.

The final step is to select the test environment and save the test. You can preview the test by clicking the Preview Run button. This will run the script while displaying the device screen.

5.13.4 Viewing the Schedule Calendar

After you click Save and Close, the calendar view is updated with your new schedule.

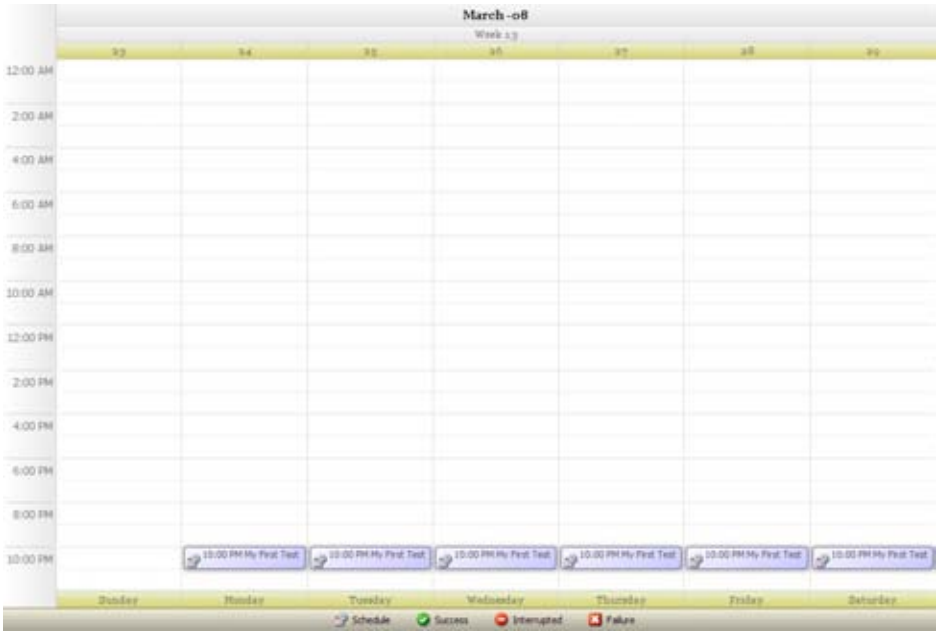


Figure 5-57 Calendar View

In this case, your test is set to run at 10:00 PM daily until you remove the test from the Schedule.

5.14 Viewing Scheduled Test Suite Results

To view Test Results, click the Test Results tab in the *MyDeviceAnywhere* web site. Test Results view has a number of links that let you drill down into the results of each test and each step in a test.

- Click the View Details link associated with your Test Suite results to view a break-down of the individual test script results. In the detailed report, the result for each test script is displayed in a row in the Test Suite Scripts table.
- For test scripts that show a Failure in the Results column, you can drill in to view a detailed log, including screen shots, of the entire script execution. Click on a Failure link to see the step-by-step results.

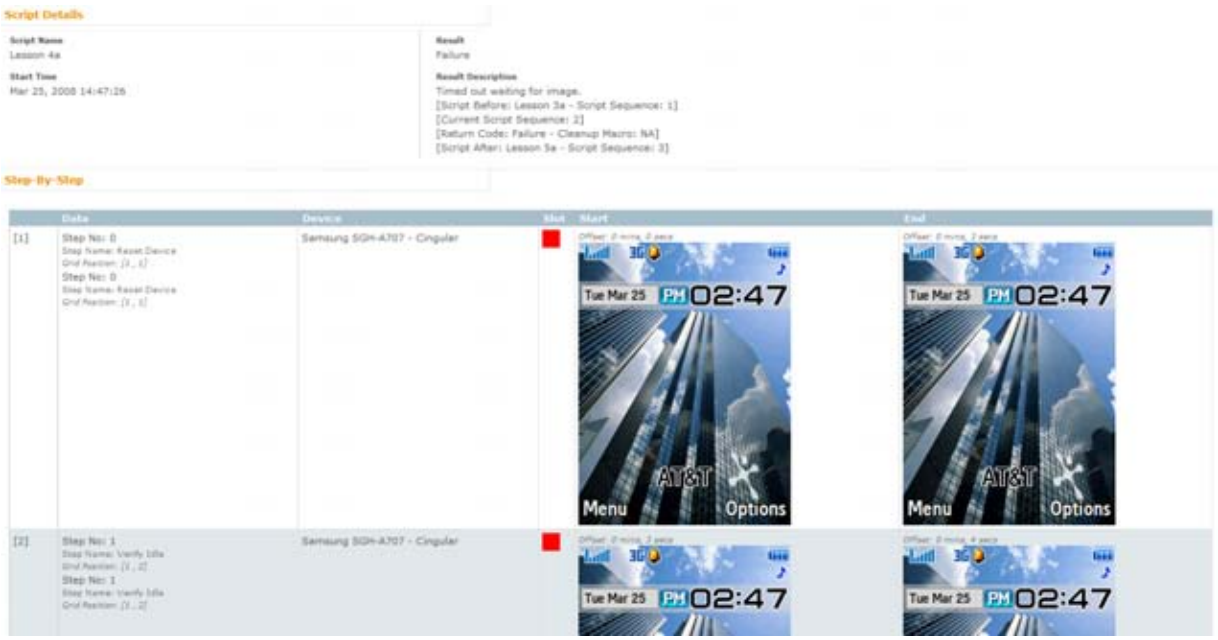


Figure 5-58 Detailed Log with Images

5.15 Dashboard

The Dashboard is a part of *DeviceAnywhere Studio* that enables you to view the inner workings of the *DeviceAnywhere Pro* Servers. The Dashboard gives you full visibility into what these servers are doing, as well as views into what any devices currently being controlled by these servers are doing.

Figure 5-59 illustrates the Dashboard view. In the Explorer, click on *DeviceAnywhere Pro* Servers to display a list of all the *DeviceAnywhere Pro* servers assigned to the account, along with high-level information in the workspace regarding their status.

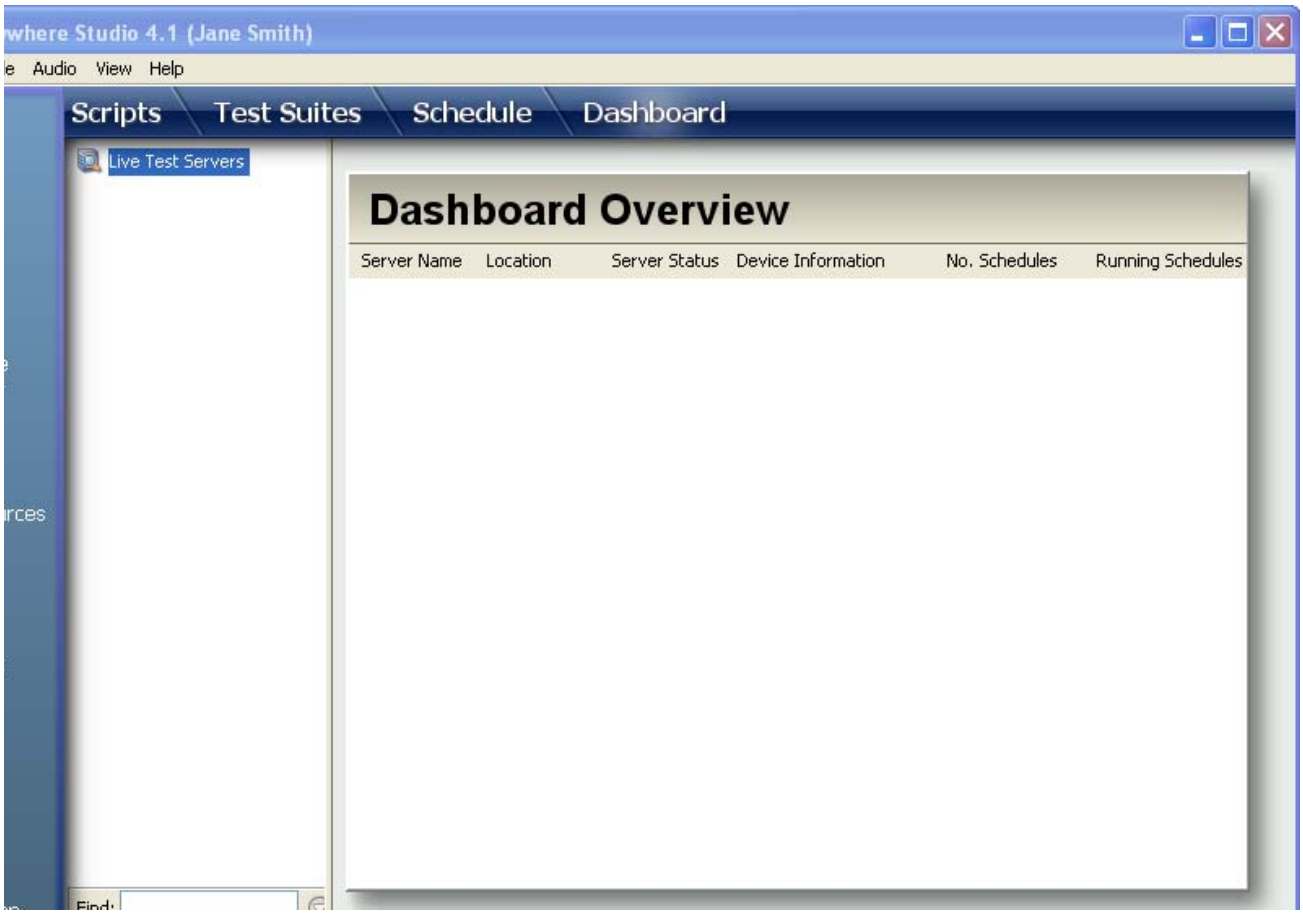


Figure 5-59 Dashboard View

5.15.1 *DeviceAnywhere Pro* Servers Menu

The *DeviceAnywhere Pro* Servers menu is opened by right clicking on *DeviceAnywhere Pro* Servers in the Explorer.

The following options are available:

- Global Reconnect—When any of the *DeviceAnywhere Pro* servers are disconnected, you can right click on *DeviceAnywhere Pro* Servers and select Global Reconnect to connect all disconnected servers.
- Configure—The Configure option displays the server configuration screen.
- Find—The Find option supports quick search of any phrase entered in the Find window.

5.15.2 Dashboard Overview

The Dashboard Overview shows a high-level view of all server tasks associated with *DeviceAnywhere Pro* servers. Only the running servers are listed here. The table has several columns as described below:

- Server Name— This column shows the name of the server.
- Location—This column shows the geographic location.
- Server Status—Running or
- Device Information shows the device bundle associated with the server.
- No. Schedules—Total number of scheduled Tests.
- Running Schedules—Tests that are currently running.

5.15.3 Preview Runs

The Preview Run is executed directly from a Test Suite. Go to the Test Suite view, create a Test Suite, and click on Preview Run in the toolbar. This action immediately executes the Test Suite on the *DeviceAnywhere Pro* servers. It creates a new temporary server with the name of the current Test Suite, and a temporary device bundle based on the current list of slot selections in the test scripts.

You can go back and analyze the local *DeviceAnywhere Pro* server results in the Dashboard while modifying the Test Suites.

5.15.4 Stop Preview Run

When you right click a temporary test server in the Dashboard that is currently running Preview Run of a Test Suite, you have the option to Stop Preview Run.

5.15.5 Delete Preview Run

Right click on *DeviceAnywhere Pro* Servers (root) in the Explorer to delete all preview runs on all *DeviceAnywhere Pro* servers.

For individual deletions, right click on any Preview Runs.

6 Monitoring

DeviceAnywhere Monitoring™ offers an environment to monitor the quality of your mobile services. Through *DeviceAnywhere Monitoring*, you can create test scripts that will test any mobile application on any device.

DeviceAnywhere Monitoring also provides the means to set and generate alerts based on user-defined performance thresholds (or SLAs), view trending reports, and schedule your tests to run at the required frequency.

DeviceAnywhere Monitoring is an enhancement of *DeviceAnywhere*. In order to use *DeviceAnywhere Monitoring*, you must have a *DeviceAnywhere Monitoring* license and a *DeviceAnywhere Monitoring* server. If you do not have either of these, you should contact the administrator of your account, or you can contact us at Sales@DeviceAnywhere.com.

Before using *DeviceAnywhere Monitoring*, you should be familiar with *DeviceAnywhere* core functionality. If you are not familiar with *DeviceAnywhere*, read the Introduction and Chapters 1 to 3 of this User Guide to get acquainted with *DeviceAnywhere*.

6.1 Monitoring Overview

DeviceAnywhere Monitoring allows you to do the following:

- Create monitoring scripts. Save sets of commands that step through tests on one or more *DeviceAnywhere* handsets.
- Configure and schedule monitoring scripts. A scheduled monitoring script is called a Production Monitor.
- Define Service Level Agreements (SLAs) and associated escalation and resolution paths to generate responses to violated SLAs (e.g. email alerts).
- View monitoring script execution in real-time.
- Launch a Production Monitor one time, providing feedback on the monitoring script before full deployment to *DeviceAnywhere Monitoring* servers.
- View reports through a web-based utility called Production Monitoring in *MyDeviceAnywhere*.

DeviceAnywhere Monitoring supports two levels of performance measurement features. These include 1) monitor pass/fail results (i.e. did the script run successfully), and 2) SLA pass/fail (i.e. did the SLA get triggered).

- Scripts can pass/fail based on the following commands:
  - The WaitImage, WaitText, WaitAudio, and Fail commands can be configured to trigger script failures.
  - Each of these types of failures can be associated with specific error types and categories for detailed reporting.

- SLAs can be defined and triggered based on three types of scenarios:
  - The success rate SLAs can be designed based on script success/failures or specific script error types.
  - The time-based performance SLAs are triggered based on the value of Toggle Timers within a script (i.e. how long a certain action took).
  - The video performance SLAs can be triggered based on the criteria you configure in the VM command within a script.

These features are included in the object-oriented scripting capabilities that distinguish *DeviceAnywhere Monitoring*.

6.1.1 Monitoring Workflow

A common workflow to get the most benefit from *DeviceAnywhere Monitoring* is provided in the following steps.

1. Create monitoring scripts in the script editor that use WaitImage, WaitText, WaitAudio, WaitEvent, and/or Fail script commands.
2. Play the monitoring scripts in *DeviceAnywhere Monitoring* for debug and verification.
3. Create Production Monitors by 1) scheduling the monitor, 2) defining SLAs, and 3) defining SLA actions.
4. Execute Monitor (run) by pressing the command in the Schedule toolbar.
5. Watch monitors running and view recent results through the *DeviceAnywhere Monitoring* Dashboard.
6. Review the results of *DeviceAnywhere Monitoring* by accessing the Production Monitoring tab in *MyDeviceAnywhere*.

6.2 Log in to *DeviceAnywhere Monitoring*

To obtain access to *DeviceAnywhere Monitoring*, first log in to *DeviceAnywhere*.

1. Launch your web browser.
2. Enter the URL: http://monitor.deviceanywhere.com
3. Enter your username and password in the Login screen.

Figure 6-1 shows the login screen for *DeviceAnywhere Monitoring*.

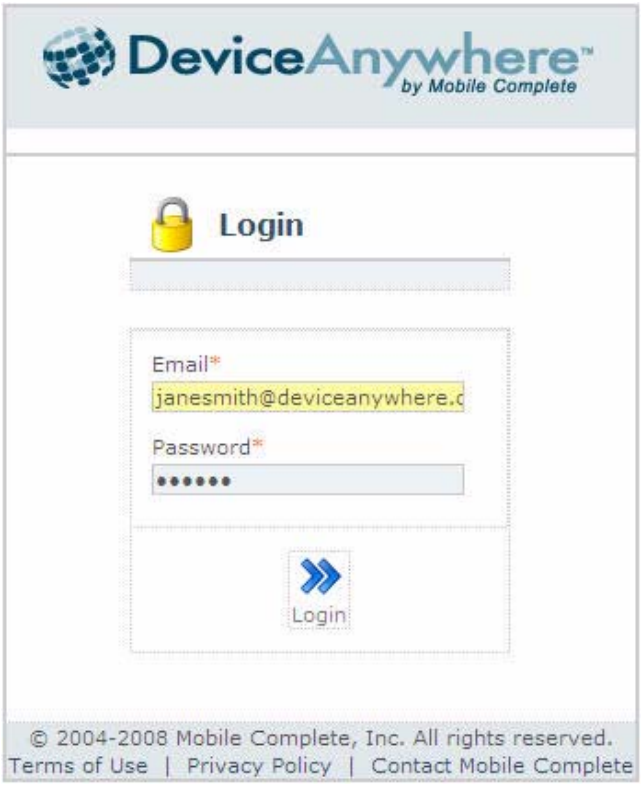


Figure 6-1 *DeviceAnywhere Monitoring* Login

Note: Log in by using the username and password provided by your Technical Account Manager. If you don't have a username and password, contact DAsupport@deviceanywhere.com.

Briefly, a summary of the login procedures is provided as follows:

1. In *MyDeviceAnywhere*, launch *DeviceAnywhere Studio* by clicking the Launch button.
2. Select Monitoring in the left Icon Bar. The Monitoring view is displayed.

For a detailed description of login procedures, refer to "Getting Started" on page 24 for further information.

6.3 Monitoring View

Figure 6-2 illustrates the Monitoring view. It includes Scripts, Schedule, Dashboard, and Preview Runs.

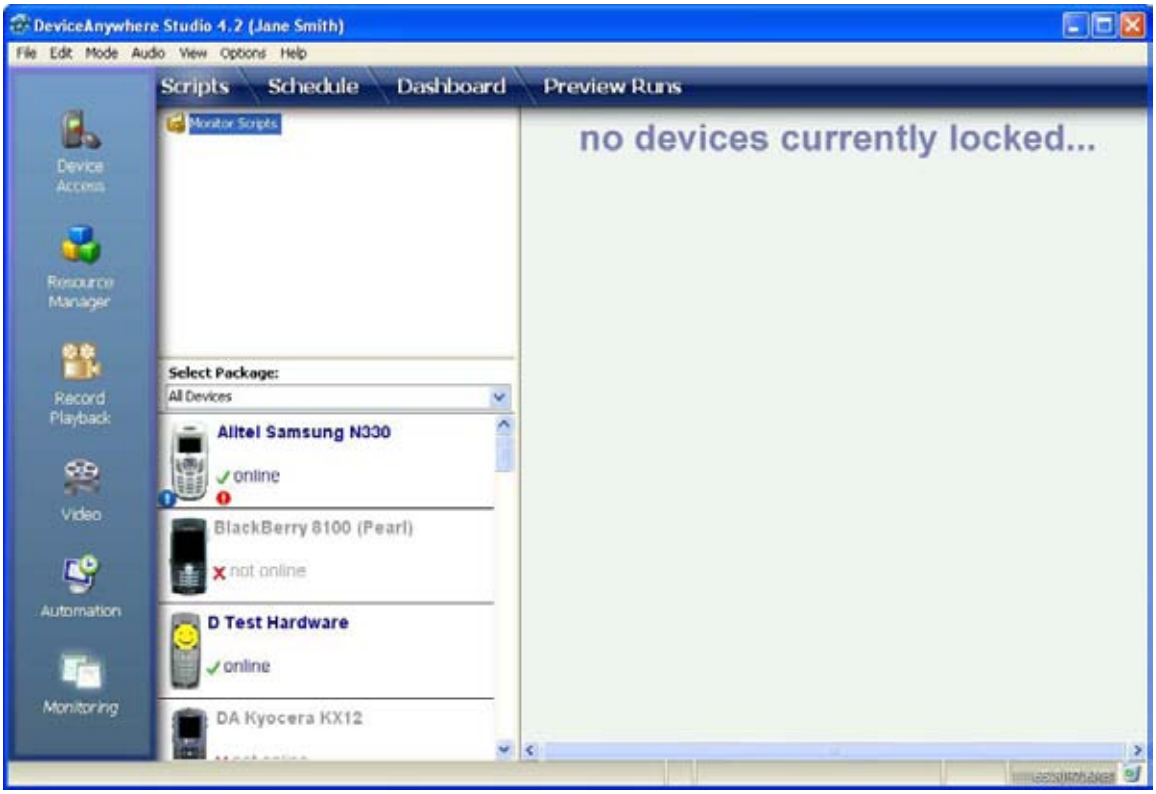


Figure 6-2 Monitoring View

A brief description of these tabs follows:

- Scripts—The Scripts tab opens the script editor, which allows you create and organize monitoring scripts for different projects.
- Schedule—The Schedule tab allows you to create a Production Monitor. The Production Monitor relies on SLAs and error management settings to detect errors and drive escalation and resolution paths. Before creating a Production Monitor, you must set up the Error Management system by selecting Options in the Menu Bar.
- Dashboard—The Dashboard allows you to view the inner workings of the past, present and future schedules for *DeviceAnywhere Monitoring* servers. The Dashboard gives you full visibility into what the *DeviceAnywhere Monitoring* servers are doing, as well as views into what any devices currently being controlled by these servers are doing.
- Preview Runs—The Preview Runs tab allows you to view the results of a Production Monitor executed one time on the *DeviceAnywhere Monitoring* servers.

DeviceAnywhere Monitoring is context sensitive. As you create folders and add scripts, the Menu Bar above the Monitoring view is expanded to include menus that support Production Monitors and monitoring scripts.

The Menu Bar in the Monitoring view provides the following menus:

- File—The file menu allows you to navigate directly to your *MyDeviceAnywhere* account.
- Edit—The Edit menu supports cut and paste functions. *DeviceAnywhere Studio* also supports standard keyboard shortcuts, such as Control+C and Control+V.
- Mode—The Mode menu allows you to set the Key mode for an entry in the device. From this menu, you can select Alpha, Numeric, or Web modes.
- Audio—The Audio menu allows you to enable or disable the speaker and to change the size of the audio buffer.
- Script—The Script menu supports slot assignments and global slot properties. Record and Playback controls are also available, in addition to Preview Run.
- Zoom—The Zoom menu allows you to select the grid size in the script editor.
- View—The View menu allows you to switch views in *DeviceAnywhere Studio*. It has the same function as the commands on the left Icon Bar.
- Options—The Options menu provides access to Error Management features. The Settings option enables you to set three settings: 1) Ad Hoc Video SLA Wait Time, 2) Maximum number of devices on screen, and 3) Wake up warnings. It also includes the About tab for a brief description of DeviceAnywhere Studio.
- Help—The Help menu provides access to the User Guide for this product and the latest Release Notes. It also supports many other links.

Refer to “DeviceAnywhere Studio Interface” on page 20 for more details on the *DeviceAnywhere Studio* interface.

6.4 Get Ready to Build a Monitoring Script

6.4.1 Use a Project

All monitoring scripts are organized within projects. The first time you use *DeviceAnywhere Monitoring*, a default project called DA-Project is available to save scripts.

Only administrators are able to set up new Projects and assign Users. You should receive a Project assignment at the same time you receive a username and password to log into your *DeviceAnywhere Studio* account.

When you don’t see your Project in the Explorer, contact your administrator. For more information on administrative tools, refer to “MyDeviceAnywhere” on page 279.

Figure 6-3 illustrates a new Project called Projects created under the Monitor Scripts (root).

Figure 6-3 Projects created under Monitor Scripts

6.4.2 Create a Folder within a Project

Folders are used to organize monitoring scripts within a project.

How to Create a Folder:

1. Click on your Project to display Monitor Scripts.
2. Right click on the Monitor Scripts. The File menu appears.
3. Select New Folder.
4. Enter the name of the folder. In this case, Check SMS Timing

Figure 6-4 illustrates the steps to create a folder to store your scripts.

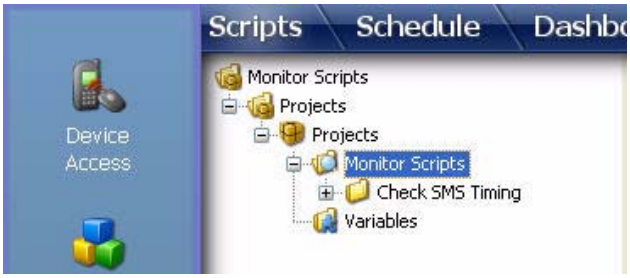


Figure 6-4 Create a Folder for Monitor Scripts

6.4.3 Create a Monitoring Script

Now that you have created a folder to work in, it's time to create your first monitoring script.

How to Create a Script:

1. Right click on the folder. The File menu appears.
2. Select New Script.
3. Enter the name of the script. In this case, SMS Timing.

The new script is now open in the script editor. One red target is shown at the top of the script. The red target indicates that your script is empty: no key presses have been recorded, and no script commands have been inserted into the script.

Generally, monitoring scripts are named for the network activity being monitored.

Figure 6-5 shows the empty script in the script editor.

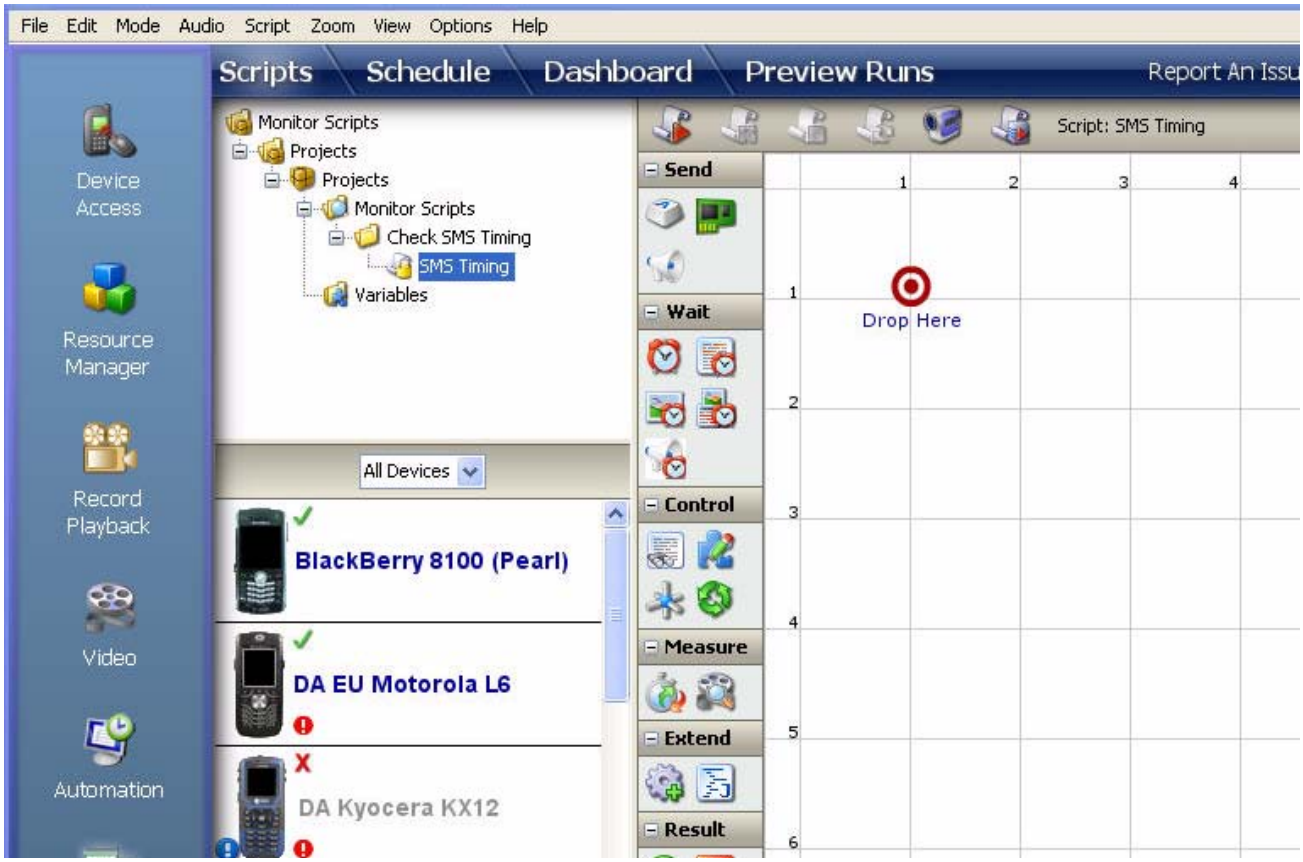


Figure 6-5 Empty Script in the Script Editor

6.4.4 Add a Device to the Script

You must add a device to the monitoring script before recording a script. In *DeviceAnywhere Monitoring*, more than one device can be added to a script. For one script, you can add up to three devices. When you add a device, the device is displayed in the far right panel of the screen.

The procedure below adds a device to the monitoring script.

How to Add a Device to a Script:

1. Right click on a device in the Device list to add the device that you want to use.
2. Select Add to Script.

When you want to remove a device from the script, right click on the device. The Device menu is displayed. This time select Remove from script.

Figure 6-6 shows the Device menu with Add to Script selected.

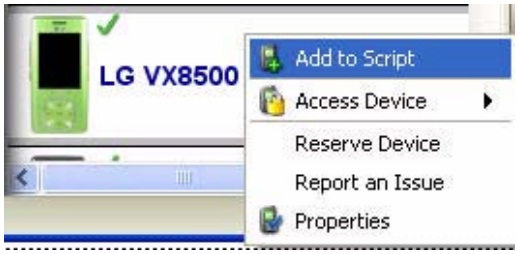


Figure 6-6 Add to Script

6.4.5 Access and Acquire a Device

Before you can create a test script, you must both access a device and acquire it.

How to Access and Acquire a Device:

1. Right click on the device in the far right panel. The Device menu appears.
2. Select Access Device.
3. Select Acquire Device.

Figure 6-7 shows the Acquire Device menu displayed.

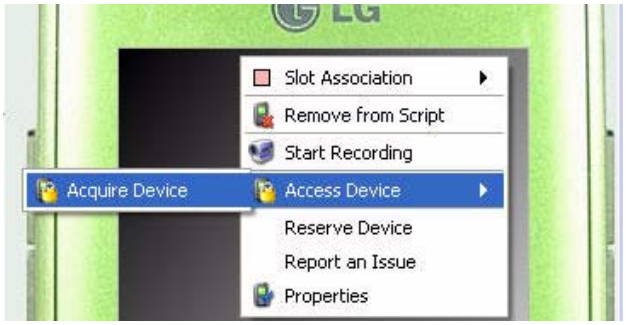


Figure 6-7 Acquire Device Menu

Note: Before you start recording make sure that the device is awake and accepting input. Most devices go into a power save mode after a certain amount of inactivity. This amount of time varies per device.

6.5 Multi-Device Scripts

Each script must be associated with at least one device. Multiple devices can also be associated to a single script to allow you to do the following:

- Run the same script across similar devices.
- Set up for tests that require more than one device (e.g., sending a text message from one device to another).

Device Bundles

Device bundles enable you to schedule a script to run across multiple sets of devices when using the automated scheduling in the Schedule tab. These device bundles are made from the available set of devices (i.e., those devices that have been added to the script). Utilizing device bundles allows you to easily schedule the running of a script across multiple sets of devices. For further information, refer to “Create a Device Bundle” on page 162.

Assigning Slots

The first device that you add to a script is assigned to Slot 1 by default. All other devices must be assigned to a slot through the Assign Slot command in the Menu Bar. For example, if you want to send a message from one device to another, you can add two devices to a script and assign the devices to slots. Each command in the script is assigned a slot. Then a single script can be run to send a message from one device to the other. For further information, refer to “Create a Device Bundle” on page 162.

Each slot is a different color to differentiate the devices. Up to 10 slots can be associated with one script (i.e., you can assign up to 10 devices to a script).

Global Allocation

TBD

6.6 Scheduler

Figure 6-8 illustrates the Scheduler view. In this view, you create Production Monitors. A Production Monitor is a monitoring script that has been scheduled.

The Production Monitor toolbar provides the commands to manage Production Monitors, such as save, run, stop, and delete Production Monitors.

The workspace has four windows that allow you to configure the parameters of a Production Monitor in the following categories:

- Monitoring Script and Device Bundle
- Execution Frequency
- Service Level Agreements (SLA) Policies
- Action Paths

Figure 6-8 illustrates the Scheduler view.

Figure 6-8 Scheduler View

In this view, the Production Monitor in the Explorer named SMS Timing is running. The monitoring script being executed is shown in the Monitor script window. In this case, it's SMS Timing.

DeviceAnywhere Monitoring is context sensitive. The commands in the toolbar change based on the status of the Production Monitor. Since a monitor is running, click Stop Monitor to stop executing the Production Monitor and the SMS Timing script.

6.6.1 Creating a Production Monitor

How to Create a Production Monitor:

1. Right click on your Project in the Explorer.
2. Select New Monitor from the drop-down menu.
3. Enter the name of the Production Monitor.

The new monitor appears in the Explorer.

6.6.2 Monitor Status

The status of a Production Monitors is indicated through an icon in the Explorer.

- A magnifying glass on the Production Monitor icon indicates that the Production Monitor is running.
- A red stop on the Production Monitor icon indicates that the Production Monitor has stopped.

6.6.3 Open the Production Monitor Menu

How to Open the Production Monitor Menu:

To open the Production Monitor menu, right click on the monitor in the Explorer.

The Production Monitor menu has the following options.

- Rename Monitor—Rename Monitor lets you modify the name of a Production Monitor.
- Save Monitor—Save the Production Monitor settings.
- Stop Monitor—Stop the execution of a monitoring script for the selected Production Monitor.
- Preview Run—Run the monitoring script for the selected Production Monitor.
- Jump to Dashboard—This option immediately navigates to the Dashboard view.
- Discard Changes—Discard Changes lets you get rid of all changes in the selected Production Monitor.
- Delete Monitor—Delete Monitor removes the Production Monitor from the Explorer.
- Properties—This option opens the Monitor Properties screen. This screen provides file information, such as author and date that the file was created and last modified.

Figure 6-9 shows the Production Monitor menu.

Figure 6-9 Production Monitor Menu

The following procedures describe how to perform simple tasks.

Rename a Monitor

How to Rename a Monitor:

1. Right click on the monitor in the Explorer. The Production Monitor menu appears.
2. Select Rename Monitor. The file name is open for editing.

Save a Monitor

How to Save a Monitor:

1. Right click on the monitor in the Explorer. The Production Monitor menu appears.
2. Select Save Monitor. The Production Monitor settings are saved.

6.6.4 Monitor Script and Device Bundle

In the following sections, you configure the Production Monitor.

Figure 6-10 shows the top window in the Scheduler view.

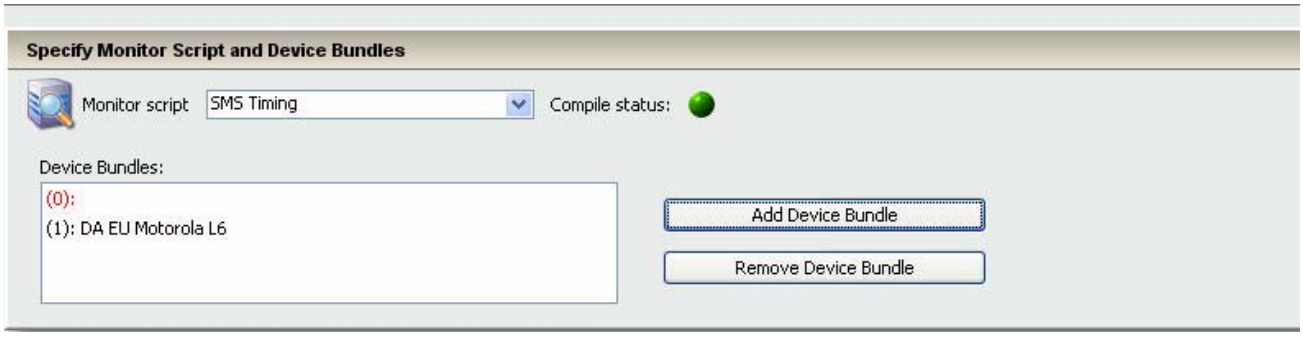


Figure 6-10 Monitor Script and Device Bundles

This window is used to specify the following settings:

- The monitoring script to run with the Production Monitor. Click on the arrow and select a script from the drop-down menu.
- The device(s) in the Device Bundle

The Compile status indicator shows that the monitoring script has been successfully compiled and is ready for execution.

The Monitor script drop-down list shows all the monitoring scripts available. You can choose only one monitoring script per Production Monitor.

6.6.5 Create a Device Bundle

When you select a monitoring script from the drop-down menu, you already have a number of devices added to the script. However, you must specify the devices explicitly to be used in the Production Monitor.

This procedure uses the Script Slot Configuration screen to assign devices to a slot and then add the devices to the Production Monitor.

Figure 6-11 shows the Script Slot Configuration screen with the DA EU Motorola L6 device assigned to the first slot.

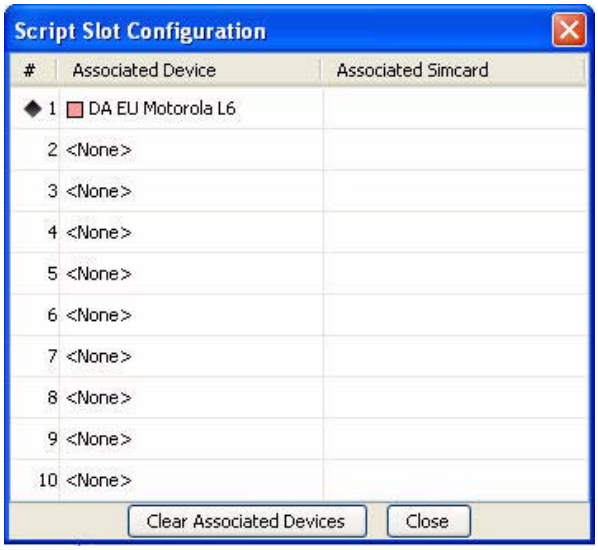


Figure 6-11 Create a Device Bundle

How to Create a Device Bundle:

1. To add a device to the Device Bundle, click the Add [to] Device Bundle button. The Script Slot Configuration screen appears.
2. Place your cursor in the first line of the screen under Associated Device. This action opens a drop-down menu. All the devices added to the monitoring script are represented.
3. Select a device for the first slot.
4. Repeat steps 2 and 3 for all devices you'd like to add to the Device Bundle.
5. Click Close.

All the devices selected for the Production Monitor are now listed in the Device Bundles window.

6.6.6 Execution Frequency

Figure 6-12 illustrates the timing parameters to schedule the Production Monitor.

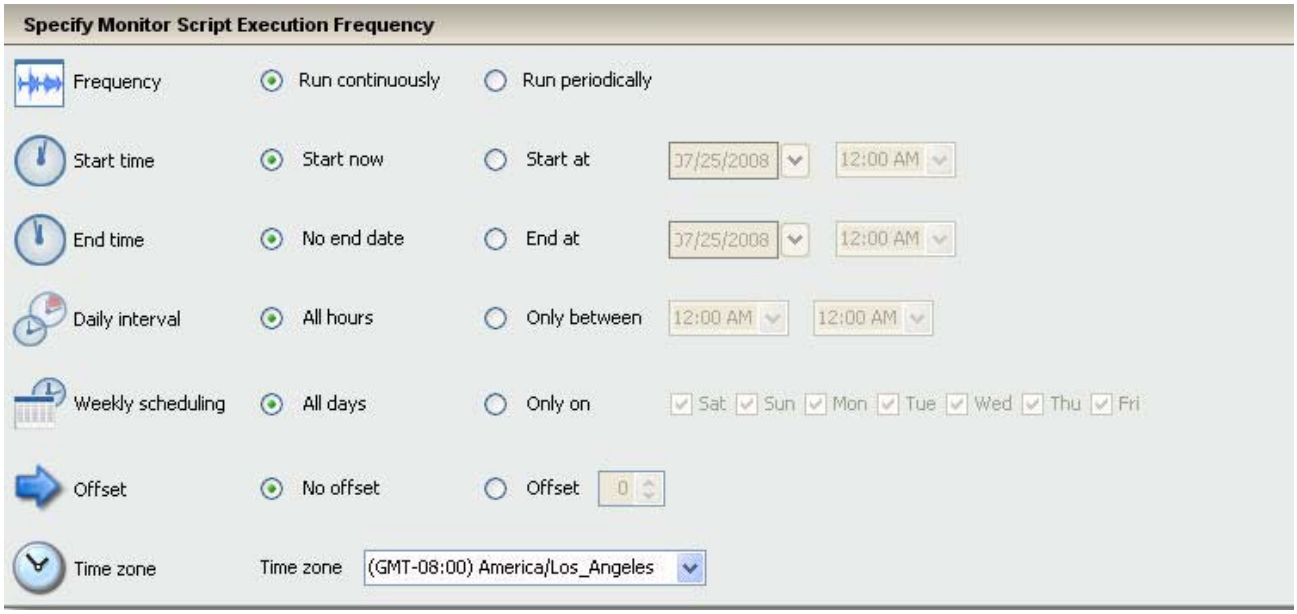


Figure 6-12 Monitor Script Execution Frequency

The following settings are configured:

- Frequency—Run Continuously: Enables your script to run throughout the defined time period; Run Periodically: Runs your script intermittently. You can define the pauses between scripts in the drop-down list. This option is only available when the Run Periodically option is selected.
- Start Time—Allows you to define the starting point of your monitoring time period. Your monitoring script starts running at the time you set here.
- End Time—Allows you to define the end of your monitoring time period. Your monitoring script does not run after the time you set here.
- Daily Interval—Allows you to define a set number of hours per day to run your monitoring script.
- Weekly Scheduling—Allows you to schedule tests on specified days of the week.
- Offset—Defines a time offset from the hour to run your monitoring script.
- Time Zone—Defines the time zone based on the location of the *DeviceAnywhere Monitoring* servers.

6.6.7 Service Level Agreements

Service Level Agreements (SLAs) that you have defined for the Production Monitor appear in the SLA Policies window.

Figure 6-13 illustrates the SLA Policies window in the Schedule view.

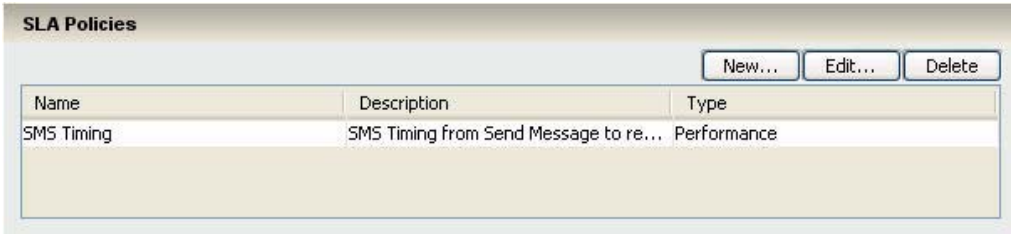


Figure 6-13 SLA Policies

The following sections describe in detail how to set up Error Management and create new SLAs for a Production Monitor.

6.6.8 SLA Overview

Service Level Agreements (SLAs) are performance metrics used to gauge the performance of network and mobile applications. These metrics determine whether your service is within the acceptable range.

SLAs are an important part of Production Monitoring. When a Production Monitor is executed, the SLAs in the Production Monitor are compared against the run-time test results of your monitoring script in order to measure network or mobile application performance on an on-going basis.

SLAs are defined with several violation rules (from low to high violation). When a SLA violation occurs, *DeviceAnywhere Monitoring* can be programmed to trigger an error message associated with a specific error type.

SLAs are tightly integrated into the error management system. Error management allows you to design custom error types that pinpoint where errors occur within your network or mobile application. SLAs are identified by the error type, category, and subcategory. This data provides a high-level of granularity in error reporting and handling.

In addition, when a SLA rule violation occurs, *DeviceAnywhere Monitoring* can trigger an appropriate escalation based on the severity of the violation. The escalation path includes an option to rerun the script, and/or to send emails and an SNMP alert message. *DeviceAnywhere Monitoring* also supports a resolution for each SLA violation.

Error management, escalation, and resolution must be set up before you can define the violation rules for an SLA. For this reason, the next section describes Error Management and the parameters that are selected when you add a new SLA.

6.6.9 Error Management

Error management is accessible by selecting Error from the Options menu in the Menu Bar. Error management has two tabs: Error Types, and Error Categories. Each tab allows you to define and manage error types and error categories accordingly.

A brief description of error type and error category is provided below:

- Error Type—An error type is a name that you create to pinpoint an error in your network or mobile application in the context of a monitoring script. The WaitImage, WaitText, WaitAudio, WaitEvent, and Fail commands in your monitoring scripts are configured with an error type on the Timeout tab in the Properties screen. In this way, the checkpoints in a monitoring script are integrated into the error management system.
- Error Category and Subcategory—An error category or subcategory is a way to organize error types under broad headings and subheadings.The WaitImage, WaitText, WaitAudio, WaitEvent, and Fail commands in your monitoring scripts are configured with an error category and subcategory on the Timeout tab in the Properties screen similar to error types.

Error types, categories, and subcategories are identified through the Timeout tab of six script commands:

- SendKeys with WaitImage checked
- WaitImage
- WaitText
- WaitAudio
- WaitEvent
- Fail

Since error types are identified by error category and subcategory, you must set up error category and subcategory first before setting up error types. Thus, the following section describes how to set up error categories.

6.6.10 Error Categories

An error category and/or subcategory serves as a filter. For example, a network provider creates a monitoring script that tests the availability of Yahoo! IM. In this script, test steps are executed to access the network, select the Yahoo! IM application, write a message, and click Send. A WaitImage is inserted into the script as a checkpoint to test loading the Yahoo! IM application.

On the Timeout tab of the WaitImage, you select the error type, category, and subcategory relevant to loading the IM application. The error type, category, and subcategory that you select is entirely dependent on the custom error management system that you create in *DeviceAnywhere Monitoring*.

In this example, the network provider creates one error category called “Network Failure,” which qualifies accessing the network. Then, the network provider creates another error category called “Third-party Application Failure” and a subcategory called “IM.” This category qualifies the Yahoo! IM feature. If Yahoo! IM fails to load, then the error type, category, and subcategory is reported in Production Monitoring test results.

How to Create an Error Category:

1. In Error Management, select the Error Category tab.
2. Click New above the Error Category window. The Add New Category dialog box appears.
3. Enter the Name of the Category.
4. Select a Category or Subcategory indicator.
5. In the Description field, enter notes that describe the error category.
6. Click Save.

Figure 6-14 shows the Add New Category screen. In this example, a new category called “Third-Party Application Failure” is defined.

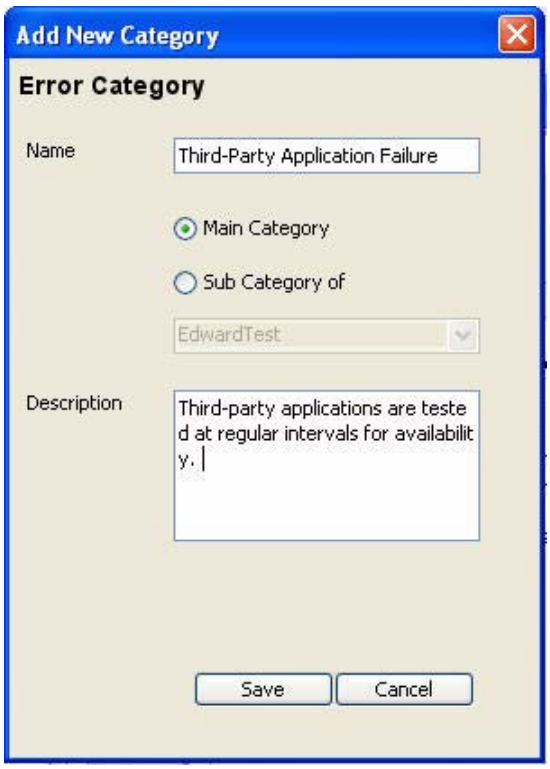


Figure 6-14 Add New Error Category

6.6.11 Error Types

An error type is an error message with properties generated when an error condition occurs. An error type consists of the following properties:

- Code Number—This number is an index automatically generated by the database to track error types.
- Name—This is a short descriptive name that you give to the error type.
- Description—This field allows you to describe the error type in greater detail to keep as notes.

- Category—The error category is selected from among the broad categories you created to set up the error management system. Error categories and subcategories support granularity in the error conditions that can occur and the appropriate escalation path that is triggered.
- Subcategory—The error subcategory is selected from among the subcategories you created to set up the error management system.
- Code—This field supports a numeric code recognized by software applications.
- Severity—This is a severity rating in *DeviceAnywhere Monitoring*.

You define error types according to the requirements of your network, mobile application, and the monitoring scripts that you are working with.

How to Create an Error Type:

1. In the Error Type screen, select New. The Add New Error Type screen is displayed.
2. Select the Main error category from the drop-down menu.
3. Select the subcategory from the drop-down menu.
4. Enter the type for this error type.
5. Enter text that describes this error type and how it will be used.
6. Select the Code indicator when you want to assign a numeric code to the error type. The entry field is open where you type in a number. Generally, these are existing error codes recognized in your network applications.
7. Select a severity level from the drop-down menu.
8. Click Save.

Figure 6-15 illustrates the Add New Error Type screen.

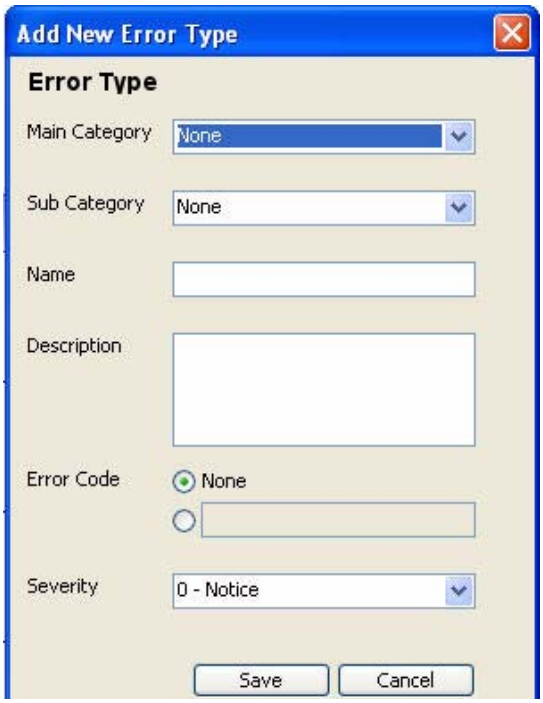


Figure 6-15 Add new Error Type

6.6.12 Severity

Error types are defined with a severity level. The severity level lets you prioritize the violation rules of an SLA to weight performance criteria. The severity settings are defined as follows:

- 0 Notice—Notification only. This message is used to send information other than an error to a distributed team.
- 1 Info—Information only. This message is used to send any message to a distributed team.
- 2 Debug—Reserved by *DeviceAnywhere Monitoring* for debug purposes.
- 3 Normal—Reserved by *DeviceAnywhere Monitoring* to indicate normal operation.
- 4 Minor—An error occurred that requires attention, but does not impair the full operation of the system.
- 5 Major—An error occurred that impairs one or more major functions of the system.
- 6 Critical—An error occurred that crashed the system or impaired most functions of the system.

6.6.13 Performing Simple Tasks with Error Types and Categories

Error management supports simple tasks like editing and deleting error categories and types. Follow the procedures below to perform these simple tasks.

Edit an Error Category

How to Edit an Error Category:

1. In Error Management, select the Error Categories tab.
2. Highlight an error category in the Manage Error Categories window.
3. Click the Edit command above the window. The Edit Error Category screen is displayed.
4. Modify the settings for the error category.
5. Click Save.

Delete an Error Type

How to Delete an Error Type:

1. In Error Management, select the Error Types tab.
2. Highlight an error type in the Manage Error Types window.
3. Click the Delete command above the window. The Delete Error type prompt screen appears.
4. Click Yes when you want to delete the error type.

Export Error Types

You can share Error Types between Projects by exporting them in an Excel spreadsheet.

How to Export Error Types:

1. In the Error Type tab, select Export above the Manage Error Types window.The Save File dialog is displayed.
2. Enter the name of the file you'd like to save.
3. Navigate to the folder where the Error Types are to be saved.
4. Click Save. The Excel file will be saved to the folder you selected.

6.6.14 SLA Types

DeviceAnywhere Monitoring supports three types of SLAs: Success Rate, Timer Performance, and Video Performance.

A brief description of these SLAs follows:

- Success Rate—This type of SLA has two important parameters: 1) sample, and 2) success rate. This SLA counts the number of times your script runs to completion over the sample you've defined. For example, you can define a success rate of 80% over a sample of five. If the monitor fails more than once, then the success rate will drop below 80% and report a violation.
- Timer Performance—This type of SLA has three important parameters: 1) threshold, 2) sample, and 3) success rate. This SLA is based on the Start/Stop Timer for a Toggle Timer pair. The Start/Stop Timer value is compared to a time threshold, such as 10 seconds, to evaluate the performance of the test step in the monitoring script.
- Video Performance—This type of SLA is based on parameters such as jitter, frame rate, frame rate average, audio present, video present, and buffering that are used to monitor video performance.

6.6.15 Create an SLA

The following sections describe how to set up Success Rate, Timer Performance, and Video Performance SLAs.

The Add New SLA screen has four windows:

- Add New SLA—This window allows you to enter the name of the SLA, a brief description, and the SLA type.
- Trigger SLA—This window shows the custom error types, categories, and subcategories you programmed into *DeviceAnywhere Monitoring*. By selecting these error types, categories, and subcategories, the SLA will automatically trigger specific error messages that help you pinpoint the error condition more accurately.
- Violation—This window allows you to enter violation rules for the SLA. The Action column allows you to select an escalation path, when defined. You can also add a new escalation path by clicking the New Escalation button.
- Resolution—This window allows you to enter the violation rule that needs to be achieved in order to consider this SLA violation resolved. The Action column allows you to select a resolution path, when defined. You can also add a new resolution by clicking the New Resolution button.

Success Rate SLA

Figure 6-16 shows the Add New SLA screen. In this case, the Success Rate SLA is shown, and the Success Rate indicator is selected.

The basic layout of the Add New SLA screen is similar for Timer Performance and Video Performance SLAs.

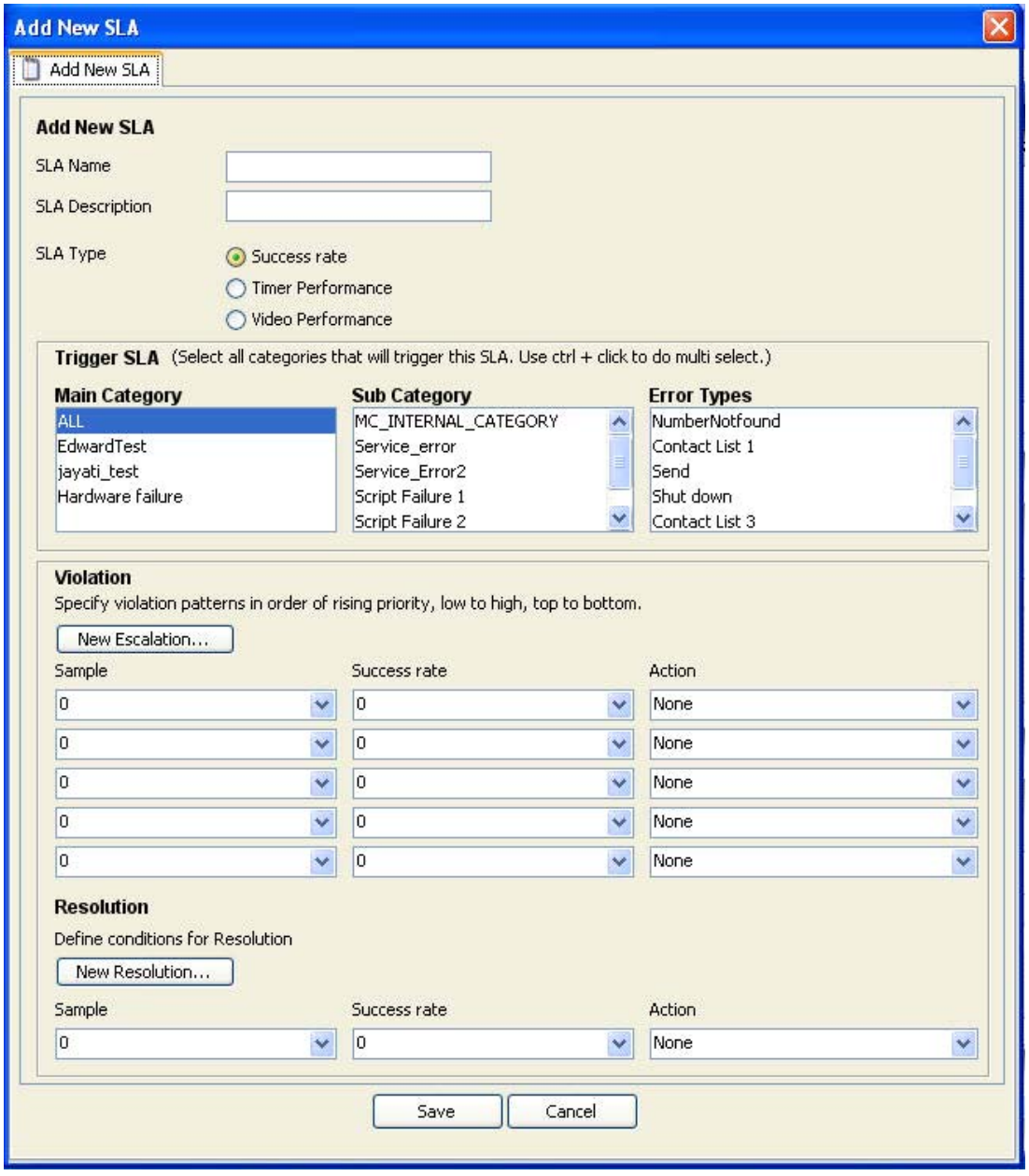


Figure 6-16 Add New SLA for Success Rate

How to Create a Success Rate SLA:

1. Click the Schedule tab in the Monitoring view.
2. Scroll down to the SLA Policies window.
3. Click New in the top right corner of the window. The Add New SLA screen is displayed.
4. Enter the name of the SLA.
5. Enter a description for the SLA that will help you use this SLA in your monitoring scripts later on.
6. Check the SLA indicator. In this case, check Success Rate.
7. In the Trigger SLA window, highlight one or more error types, categories, and subcategories that will trigger this SLA.
8. In the Violation window, enter all the violation rules that represent possible error scenarios in increasing magnitude.
9. In the Violation window, select the appropriate escalation path for each violation rule.
10. In the Resolution window, enter the violation rule that indicates the error condition is resolved.
11. In the Resolution window, select the resolution for this SLA.
12. Click Save. The new SLA will appear in the SLA Policies window.

Timer Performance SLA

A Timer Performance SLA measures timing performance for test steps or an entire monitoring script by using Toggle Timers. For example, Toggle Timers are used when you want to test the time elapsed to send an SMS message, clear the cache, or load a video file on a network and/or device.

To set up timing in a monitoring script, drag a Toggle Timer into the script and configure it to start the clock. Then, after setting up intervening steps that you want to measure time on, drag another Toggle Timer into the script and configure it to stop the clock.

Each Toggle Timer is configured with a Timer. To create a Timer, double-click the Toggle Timer. When the Properties screen is displayed, select the Timers tab. Enter a name for the timer, such as Start or Stop Send Message (where appropriate), and click the Add New Timer button.

Each Timer has a timer type. *DeviceAnywhere Monitoring* supports three timer types as described below:

- One Time: A One Time timer reports only the time measured on the previous execution. For example, if a monitoring script runs six times, a One Time timer measures the time interval six times, but only returns the last time recorded.
- Average: An Average timer returns the average time interval of all the recorded executions.
- Summation: A Summation timer returns the sum total of all the time intervals measured during the recorded executions.

A Timer Performance SLA measures performance against the Stop Timer value in your script, since it represents an elapsed time.

Figure 6-17 illustrates the Timer Performance SLA screen. In this case, the Timer Performance indicator is selected. The Add new SLA screen is similar to the Success Rate screen. Settings related to timing are included. In the Timer field, you set the Stop Timer from the drop-down menu.

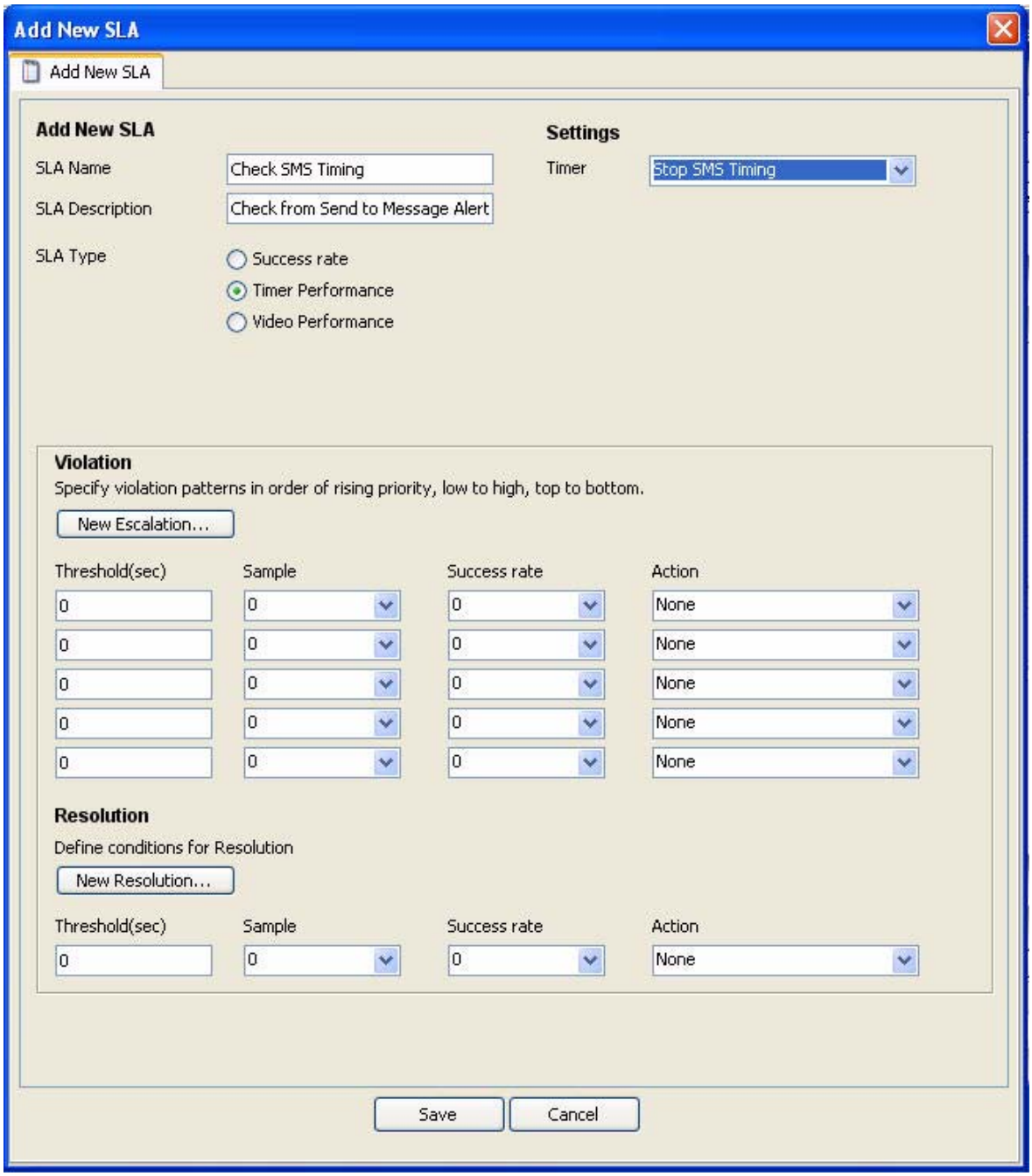


Figure 6-17 Timer Performance SLA

How to Create a Timer Performance SLA:

1. Click the Schedule tab in the Monitoring view.
2. Enter the name of the monitoring script for this Production Monitor.
3. Enter the device bundle.
4. Scroll down to the SLA Policies window.
5. Click New in the top right corner of the window. The Add New SLA screen is displayed.
6. Enter the name of the SLA. In this case, Check SMS Timing.
7. Enter a description for the SLA that will help you use this SLA in your monitoring scripts later on.
8. Check the SLA indicator. In this case, check Timer Performance.
9. In Settings, use the drop-down menu to select the Stop Timer in your script for the portion of the script under test.
10. In the Trigger SLA window, highlight one or more error types, categories, and sub-categories that will trigger this SLA.
11. In the Violation window, enter all the violation rules that represent possible error scenarios in increasing magnitude. In this case, enter the threshold in seconds, sample, success rate, and action.
12. In the Violation window, select the appropriate escalation path for each violation rule in the Action column.
13. In the Resolution window, enter the violation rule that indicates the error condition is resolved.
14. In the Resolution window, select the resolution for this SLA.
15. Click Save. The new SLA will appear in the SLA Policies window.

Video Performance SLA

A Video Performance SLA measures video performance on the network or in a mobile application by using the Video Monitor (VM) command. For more information on the Video Monitor, see “Video Monitor” on page 187.

VM commands measure video performance like frame rate, jitter, and the presence of audio and video. Like a stopwatch, a VM command needs to be started and stopped.

To set up video monitoring, drag a VM command into the script and configure it to start the video monitoring. Then, after setting up intervening steps to view a video clip, drag another VM command into the script and configure it to stop video monitoring.

The Video Monitor is able to monitor video based on the following parameters:

- Audio present
- Frame rate
- Buffering
- Frame rate average
- Audio volume
- Video present

The analysis results are based on calculations of the MC Video Quality plug-in. It analyzes video data in a general way. For example, MC Video Quality is able to detect a rapid decline in frame rate and will report errors based on the violation rules for the SLA.

Video SLA Violation Rules

Video SLA violation rules are defined relative to an optimum setting. For example, 10 frames per second may be considered an acceptable baseline for frame rate. SLAs for frame rate are expresses as a value less than or greater than the optimum setting. When the frame rate equals less than 5 frames per second, then a major error has occurred.

Figure 6-18 illustrates the Video Performance SLA screen. In this case, the Video Performance indicator is selected. These screen includes settings related to video performance.

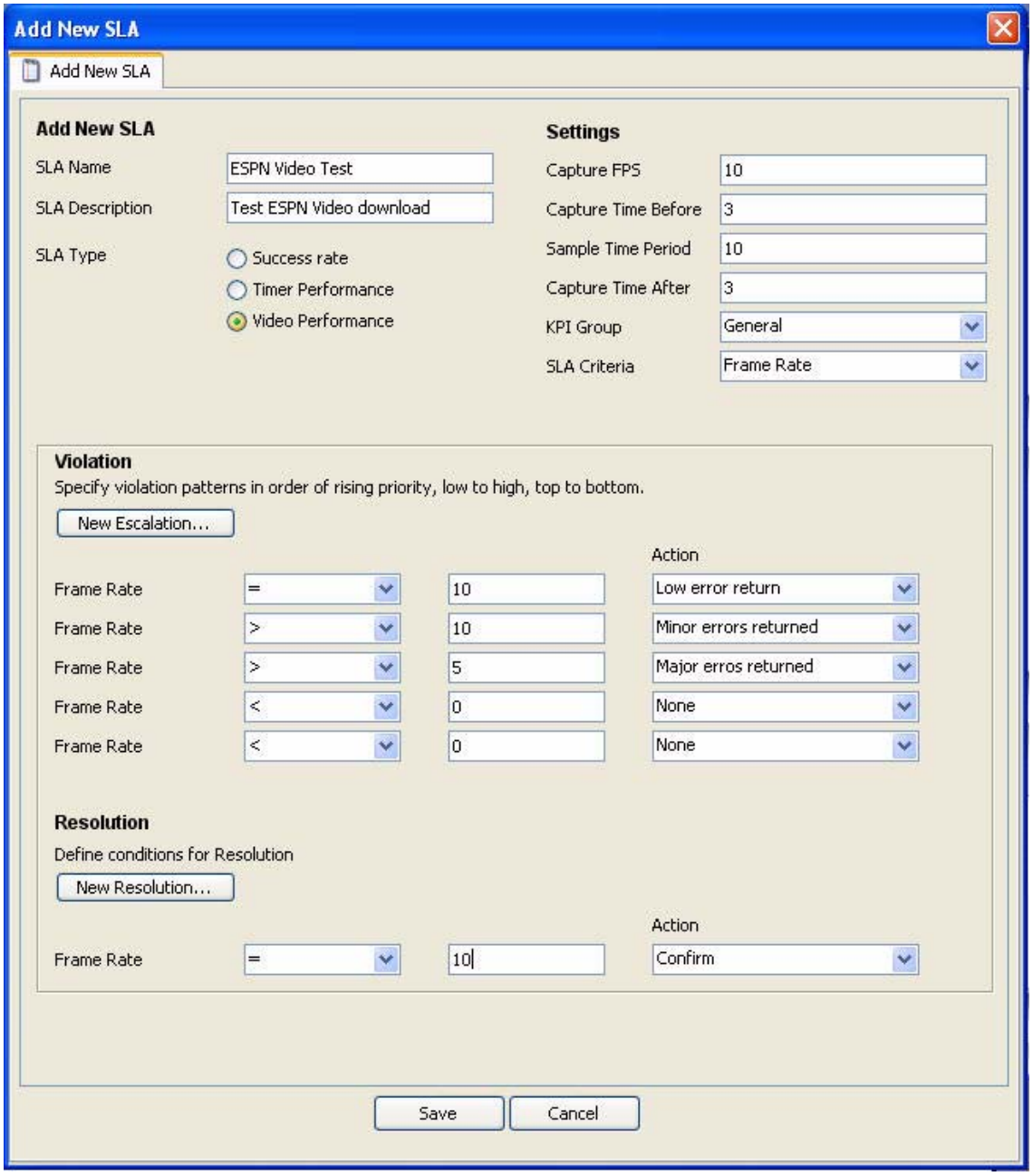


Figure 6-18 Video Performance SLA

How to Create a Video Performance SLA:

1. Click the Schedule tab in the Monitoring view.
2. Scroll down to the SLA Policies window.
3. Click New in the top right corner of the window. The Add New SLA screen is displayed.
4. Enter the name of the SLA. In this case, ESPN Video Test.
5. Enter a description for the SLA that will help you use this SLA in your monitoring scripts later on.
6. Check the SLA indicator. In this case, check Video Performance.
7. In Settings, enter the capture Frames Per Second (FPS) rate. In this case, enter 10 frames.
8. Enter the capture time before value. In this case, 3 seconds.
9. Enter the sample time period. In this case, 10 seconds.
10. Enter the capture time after value. In this case, 3 seconds.
11. Enter the KPI Group. General is the default setting. You can only select another setting when you have developed your own analysis plug-in and integrated it into the *DeviceAnywhere Monitoring* environment.
12. Select the SLA criteria that will be measured for this SLA.
13. In the Violation window, enter all the violation rules that represent possible error scenarios in increasing magnitude. For example, when frame rate drops below 5 frames per second, a major violation has occurred.
14. In the Violation window, select the appropriate escalation path for each violation rule in the Action column.
15. In the Resolution window, enter the violation rule that indicates the error condition is resolved.
16. In the Resolution window, select the resolution for this SLA in the Action column.
17. Click Save. The new SLA will appear in the SLA Policies window.

6.6.16 Manage SLA Policies

You can create, edit, and delete SLA Policies in the SLA Policies window of the Schedule view. Just click on the corresponding command in the SLA Policies window.

6.6.17 Action Path

The escalation and resolution paths that you have defined for the Production Monitor appear in the Action Path window.

Figure 6-19 illustrates the Action Path window in the Schedule view. In this case, an escalation path has been defined for the SMS Timing script. It shows three escalations: Low, Minor, and Major. Actions are scheduled based on the level of severity of the SLA violation.

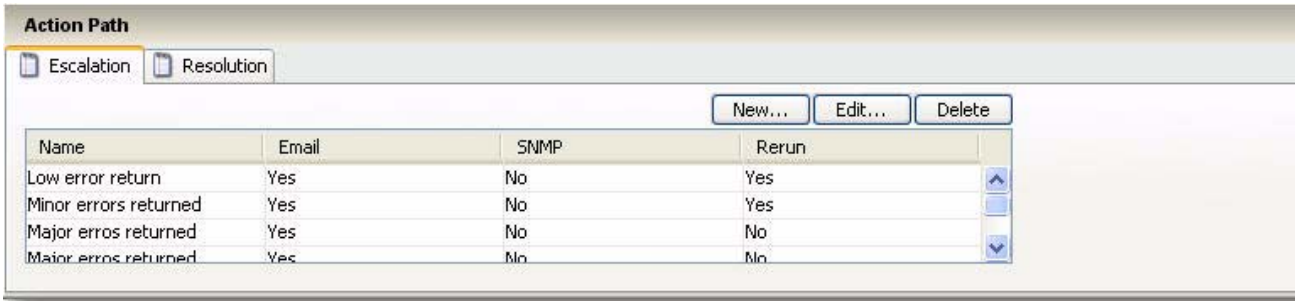


Figure 6-19 Action Path

The Action Path window has two tabs: Escalation and Resolution. Select each tab to manage and edit the escalation and resolution paths accordingly.

A brief description of escalation and resolution follows:

- Escalation is an alert, based on the severity of an SLA violation. An escalation is defined for each violation rule in an SLA policy. An escalation determines whether the script should be rerun. It also specifies the people who receive email alerts and/or one or more IP addresses that should receive SNMP alert messages.
- Resolution is the action that needs be taken when an SLA violation is resolved. A resolution specifies the people who receive email alerts with the final status of the SLA violation, or other actions to be taken, when a violation is resolved.

6.6.18 Escalation

Escalations are defined from low to high alert status. Escalation supports the following actions:

- Rerun a script
- Send alert emails to distributed team members
- Send SNMP alert messages to an IT, Data Center, or Third-Party server

These three actions can be combined in one escalation.

Figure 6-20 shows the Add Escalation Path screen.

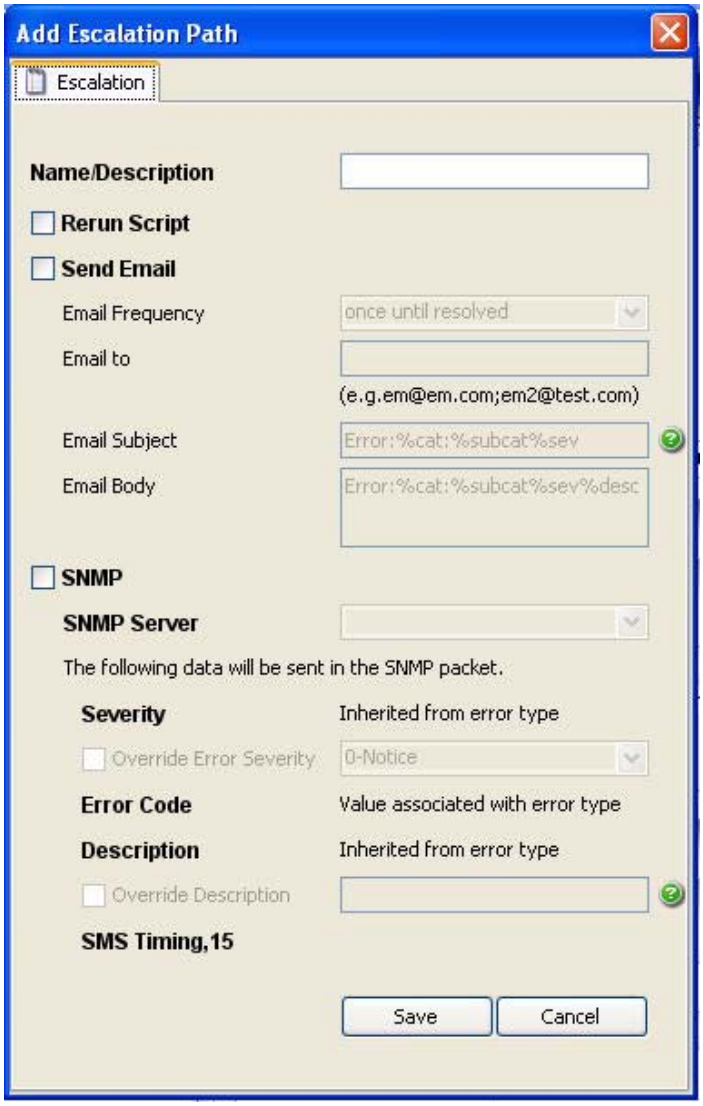


Figure 6-20 Add Escalation Path

How to Add an Escalation Path:

1. In Schedule view, scroll down to the Action Path window.
2. In the Escalation tab, select New. The Add Escalation Path screen is displayed.
3. Enter the name of the escalation path. Be sure to tailor this escalation path for the SLA violation rule that will trigger the escalation path.
4. Check Rerun script when you want to rerun the monitoring script.
5. Check Send email when you want to define a list of team members who will receive email alerts in case the SLA violation occurs and the corresponding escalation path is triggered.
6. Fill out the Email fields—Email frequency, the Email list of names, the Email subject, and the Email body, with the appropriate information.

7. Check SNMP when you want to send SNMP messages to servers internally and externally to your organization. Messages are generally sent to the IT department, Data Center, or Third-Party service providers.
8. Fill out the SNMP fields—SNMP server IP address(es), severity, error code, and description. The last three fields are inherited by the error type and filled in by default.
9. Check the override checkbox when you want to change the settings for severity and description from the default.
10. Click Save. The new escalation path appears in the Action Path window on the Escalation tab.

6.6.19 Resolution

A Resolution is an action to notify recipients of the status of a problem. It can be customized to fit your error handling practices. A resolution can be tailored to a specific error type and/or a severity level of an SLA violation. *DeviceAnywhere Monitoring* allows you to define multiple resolutions.

Resolutions are created in the Action Path window of the Schedule view.

How to Add a Resolution:

1. In Schedule view, scroll down to the Action Path window.
2. In the Resolution tab, select New. The Add Escalation Path screen is displayed.
3. Enter the name of the resolution policy. Be sure to name the policy so that you can identify the kind of SLA the resolution policy can be applied to.
4. Check Send email when you want to define a list of team members who will receive emails with a final resolution.
5. Fill out the Email fields—the Email list of names, the Email subject, and the Email body, with the appropriate information.
6. Check SNMP when you want to send SNMP messages to servers internally and externally to your organization. Messages are generally sent to the IT department, Data Center, or Third-Party service providers.
7. Fill out the SNMP fields—SNMP server IP address(es). The severity and error code are filled in by default. The severity level is for a resolution policy is always level 3. Severity level 3 is reserved by the system to report normal operation. Error code always corresponds to the last error code that triggered the SLA.
8. Enter text into the Description field when you want to keep notes on the resolution.
9. Click Save. The new resolution path appears in the Action Path window on the Resolution tab.

Figure 6-21 illustrates the Add Resolution screen.

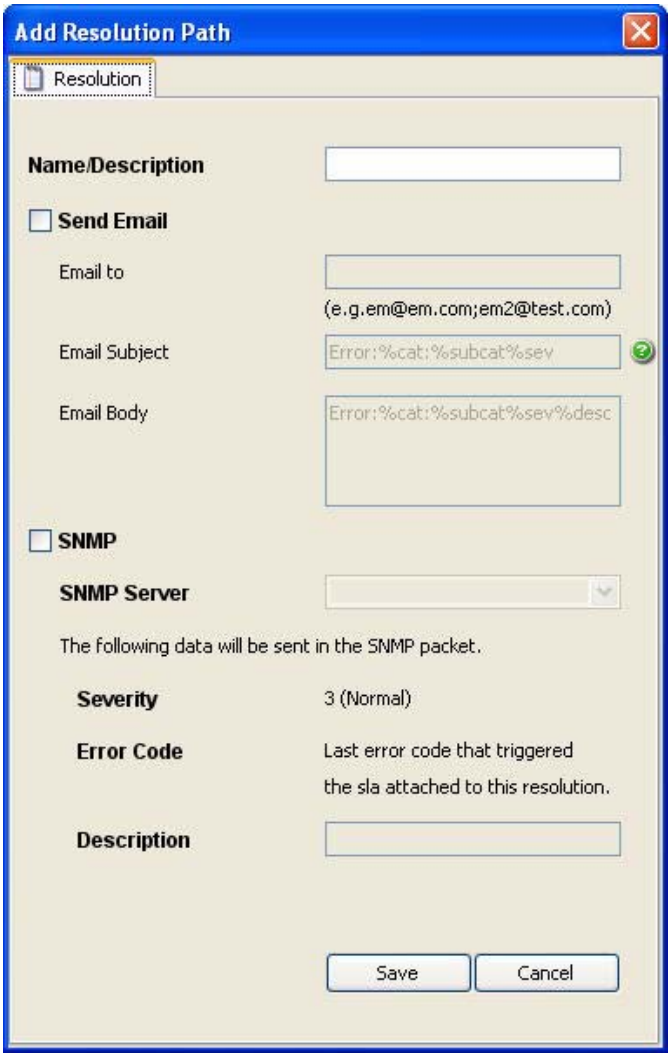


Figure 6-21 Add Resolution Path

6.6.20 Execute Production Monitor

The Execute Monitor command in the toolbar deploys the Production Monitor on the *DeviceAnywhere Monitoring* server.

6.7 Dashboard

The Dashboard allows you to view the inner workings of the past, present and future schedules for the Production Monitors that you have created. The Dashboard gives you full visibility into what the Production Monitors are doing, as well as what any devices currently being controlled by these Production Monitors are doing.

The Dashboard also shows the number of *DeviceAnywhere Monitoring* servers in your network. *DeviceAnywhere Monitoring* servers are shown with a server icon in the Explorer.

Figure 6-22 illustrates the Dashboard view. In this view, one *DeviceAnywhere Monitoring* server is shown.

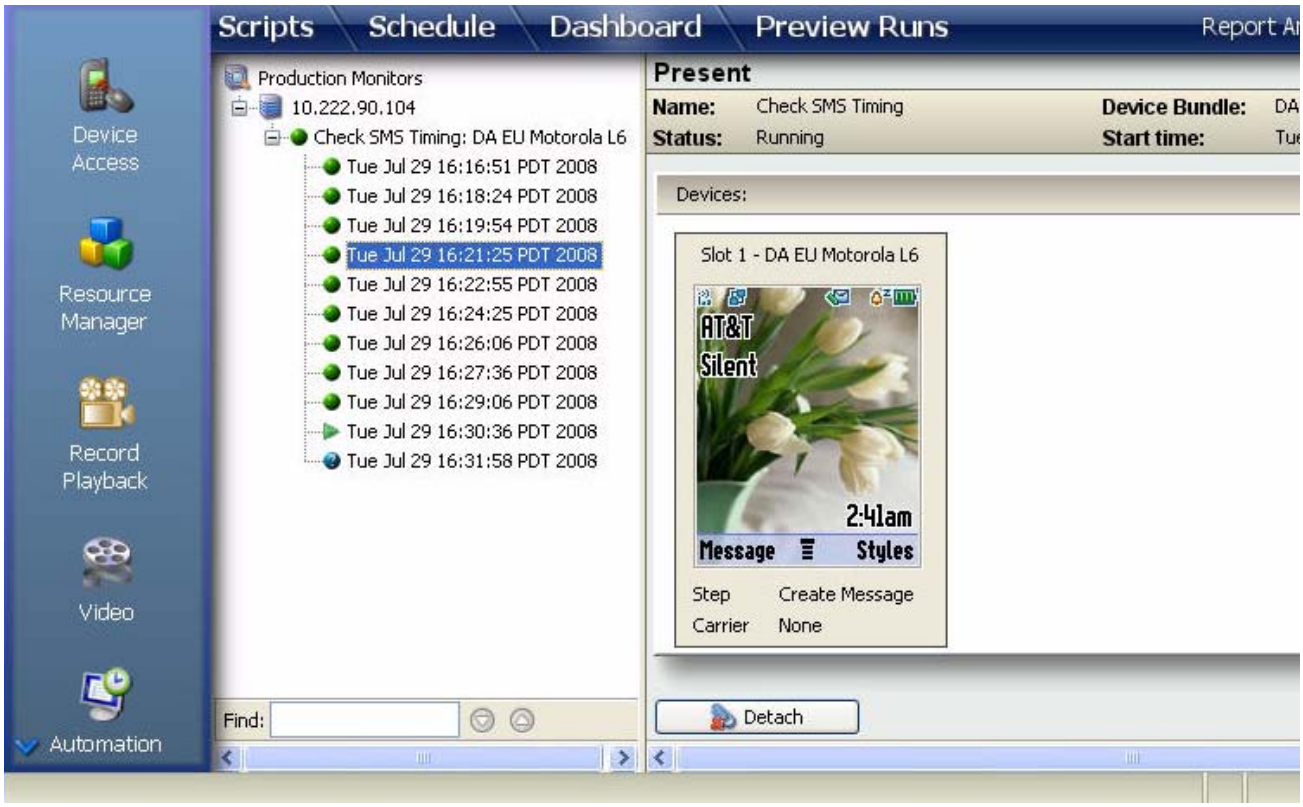


Figure 6-22 Dashboard View

In the Explorer, Production Monitors are listed. Under each monitor, a listing of each runtime log is shown. For example, the Check SMS Timing monitor ran on Tuesday, July 29.

Status indicators are shown in front of each monitor and runtime log.

- A green dot indicates a 'run' status. Ready and scheduled to run accordingly.
- A red dot indicates a Production Monitor has failed.
- A blue dot indicates currently active.
- A purple dot indicates a Preview Run completed.
- A yellow dot indicates the current Production Monitor has encountered script failures.

The workspace is further divided into three windows (not shown):

- Present—This window shows you the monitoring script that is currently running. It allows you to attach the device to the desktop. Thus, all images on the device screen for the active monitor are streamed to your desktop. On the right, a set of tabs allows you to view the metrics gathered for the current run.
- Future—This window shows scheduled time for execution and task status.1) Pause allows you to pause the monitor, 2) Run Now allows you to execute a future task immediately, 3) Start allows you to start a paused monitor (button appears when monitor is paused).
- Past—This window allows you to select previous runtime logs and view the metrics gathered for the run.

Figure 6-23 illustrates the Past window. Here you see the successful completion of a monitoring script. Click on one of the tabs on the right to view the metrics captured for the run.

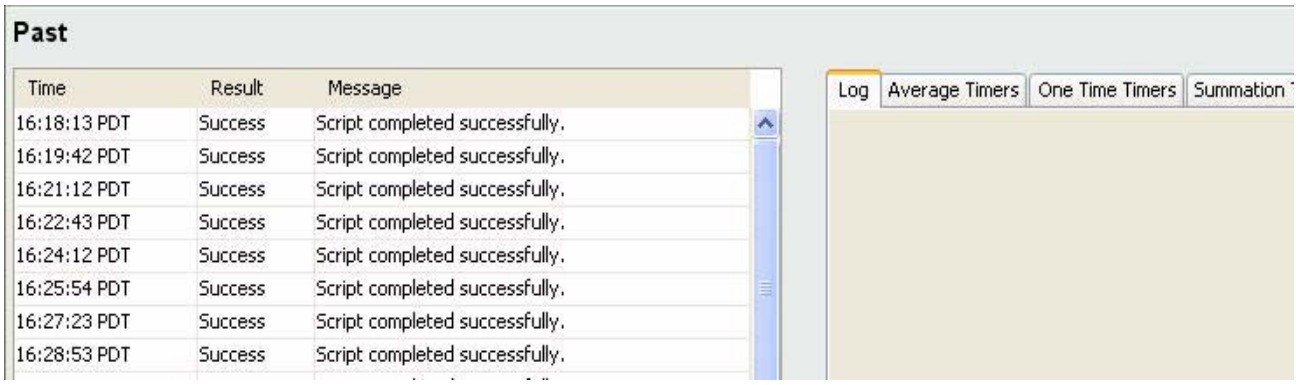


Figure 6-23 Log from Previous Runs

Four tabs provide access to metrics information:

- Average Timers
- One Time Timers
- Summation Timers
- Key Performance Indicator (KPI) Data

For further information on the Timers, refer to “Toggle Timers” on page 245.

For further information on KPI Data, refer to “Video Monitor” on page 187.

6.7.1 Global Reconnect

When any of the *DeviceAnywhere Monitoring* servers are disconnected, you can right click on Production Monitors (root), and click on Global Reconnect to connect all disconnected servers.

6.7.2 Configure

Right click on a monitor and select the Configure option. *DeviceAnywhere Monitoring* navigates immediately to the Schedule view.

6.7.3 Find

The Explorer supports a quick search of any phrase entered in the Find entry field.

6.8 Preview Runs

Preview Runs allows you to verify your monitoring scripts and the corresponding Production Monitor.

The Preview Runs command is available in the Script and Schedule views in the toolbar. You can navigate immediately to the Preview Run by right clicking a script and selecting Preview Run.

When you right click a monitor that is currently running a preview run of a monitoring script, you have the option to Stop Preview Run by selecting this command from the drop-down menu.

Figure 6-24 Preview Run View

In the Explorer, the preview runs for the Production Monitor are listed. In this case, only one Preview Run is listed.

The Preview Runs view is divided in two windows:

- Top Window—The top window shows images for the device as the script is executed. When the Preview Run is complete, the Video Control bar appears below the device. Use these controls to scan through the recorded images.
- Bottom Window—The bottom window contains the runtime log for the preview run.

The bottom window has four tabs that provide metrics information as follows:

- Average Timers
- One Time Timers
- Summation Timers
- Key Performance Indicators (KPI) Data

When you select a given log entry in the bottom window, the device screen shows the corresponding frame captured during script execution in the top window.

The log provides useful information for script debugging. You can double click on these log items:

- SendKey: Shows a pop-up box with the key that was sent, the slot number and other properties.
- WaitForImage: Shows a pop-up box with the image, the slot number and other properties
- WaitForText: Shows a pop-up box with the text and the slot number.
- WaitForEvent: Shows a pop-up box with the text or image and the slot number used.
- NavigateTo: Shows a pop-up box with the text to send, the image and the slot number used.

For further information on the Timers, refer to “DeviceAnywhere Script Commands” on page 223.

For further information on KPI Data, refer to “Video Monitor” on page 187.

6.9 Video Monitor

DeviceAnywhere Monitoring supports a tool for testing and monitoring video on mobile devices. The Video Monitor is used in two ways:

- It can be used as a stand-alone feature to perform ad hoc video/audio testing.
- It is used in the background to perform statistical analysis in recurring monitoring scripts. The parameters you set to test video are selected through the Add New SLA screen when Video Performance is selected. For more information, see "Create an SLA" on page 170.

When you want to monitor video performance on a live network, you use the VM script command in a recurring monitoring script. For further information on the Production Monitor, refer to "Scheduler" on page 159.

The Video Monitor is available by clicking Video in the left Icon bar.

6.9.1 Video Monitor Overview

Figure 6-25 illustrates the Video Monitor view. This view includes two tabs for KPI Config and Video Monitor.

- In the KPI Config tab, the General KPI group is available for ad hoc video monitoring.
- In the Video Monitor tab, you turn on the Video Monitor and interact with a device.

The Explorer shows the General KPI group and a New KPI group as an example.

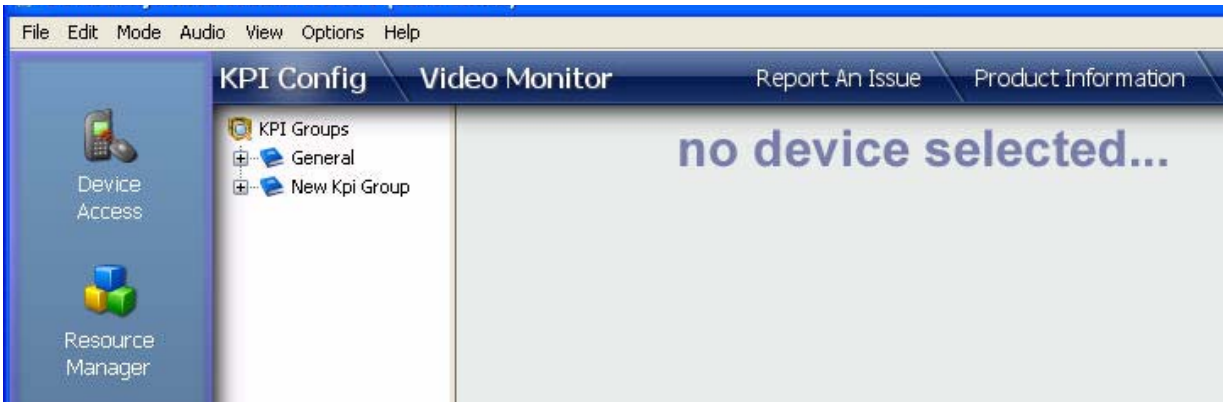


Figure 6-25 Video Monitor View

6.9.2 General KPI Group

KPI means Key Performance Indicators. The General KPI group is the default used to support *DeviceAnywhere Monitoring* SLAs based on Video Performance.

The General KPI group is associated to the MC Video Quality analysis plug-in. MC Video Quality is a tool to calculate frame rate, jitter, and so on.

You have the option to create a KPI group and associate it to *DeviceAnywhere Monitoring*. However, this advanced topic is not covered here. If you want to create a new KPI group and analysis

plug-in and then integrate it into *DeviceAnywhere Monitoring*, contact Support@deviceanywhere.com for detailed information.

6.9.3 Video Monitor Ad Hoc Testing

Typically, you perform video testing and monitoring in an ad hoc manner in the Video Monitoring view. In this first step, you observe system response and the results obtained per device. This information helps you obtain a realistic threshold or measurement for video performance SLAs.

Video Monitor has the ability to perform full-frame video capture and analysis at 30 frames per second. The Video Monitor supports the following parameters:

- Frame rate
- Jitter
- Frame rate Average
- Audio Volume
- Audio Present
- Video Present
- Buffering

Follow the procedures below to perform ad hoc video monitoring.

6.9.4 Add Device to KPI Group

1. Click on the KPI group in the Explorer to display the Video plugin.
2. Select the plugin with the cursor.
3. Right click on a device in the Device list. The Device menu appears.
4. Select Add to KPI Config view.
5. The device is added to the KPI group above the plugin selected in the Explorer.

The device is immediately shown in the Explorer. The device is added to the KPI Config view in the right-most panel of the screen. The device name is displayed under MC Video Quality.

6.9.5 Access and Acquire a Device

1. Right click on the device in the workspace. The Device menu appears.
2. Select Access Device.
3. Select Acquire Device.

The following settings are configured:

- Select KPI Group Parameter—This field allows you to select the specific parameter to be used for analysis.
- Enabled Checkbox—You must select this checkbox to make the KPI Group valid for use in analysis
- Start/Stop KPI Monitor—Once you have configured all the settings, start the Video Monitor. Interact with the device in the workspace. Analysis results are displayed on the screen.

To see the affect of the configuration settings, click the Enabled checkbox and click the Start KPI Monitor button.

The KPI Chart is generated. The line displayed in the moving graph corresponds to the KPI parameter selected at the top of the KPI Config tab. If any settings are changed for the selected parameter, you must restart the KPI monitor to reflect those changes. Selecting Stop KPI Monitor stops the process.

6.9.6 Video Monitor

Once you have set the parameters in the KPI Config view, you will want to monitor video performance on the device and define SLAs for video performance.

Begin by clicking on the Video Monitor tab in the Video view.

1. Right click on the All Video Monitor Views (root).
2. Select Add View to add a new video monitor.

The new view has been added to the All Video Monitor Views (root).

Figure 6-26 Add View to the Video Monitor

6.9.7 Add a Device and Acquire the Device

1. Right click on a device in the Device list. The Device menu appears.
2. Select Add to Video View. The device is added to the far right panel of the screen.
3. Right click the device in the workspace. The Device menu appears.
4. Select Access Device.
5. Select Acquire Device.

The device name is added to the All Video Monitor Views in the Explorer. You can add more than one device to the Video Monitor view.

Click on the device name, and the Video Monitor screen is displayed.

Figure 6-27 illustrates the Video Monitor view.

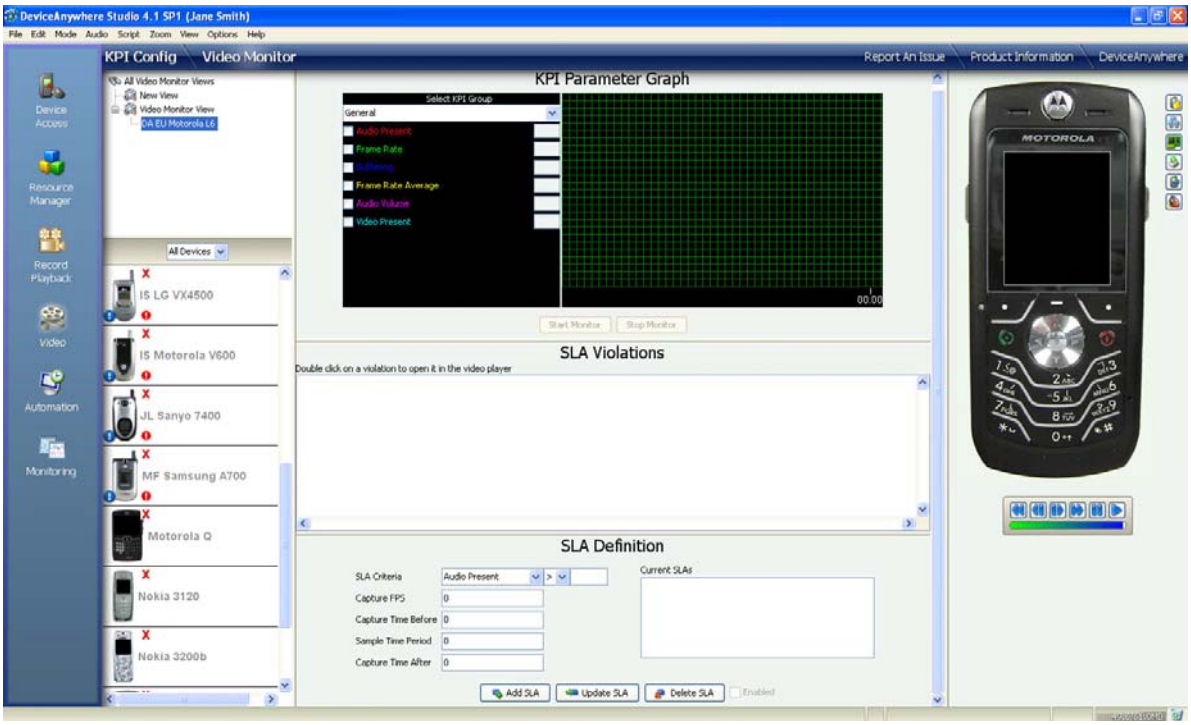


Figure 6-27 Video Monitor View

The Video Monitor view has three windows:

- KPI Parameter Graph—In this window, select the parameter for ad hoc testing. Enter the threshold in the entry field after the parameter. You can start and stop the monitor by clicking the Start Monitor and Stop Monitor buttons below the KPI Parameter graph.
- SLA Violations—This window shows the SLA violations that occurred during a test run.
- SLA Definition—This window allows you to create new SLAs for video performance and add them to the ad hoc video monitor.

You can define each individual SLA by giving values to each of the five parameters defined in the SLA Definition window. Add, Update, and Delete SLAs by clicking the three buttons below the SLA Definition window. These commands let you manage the SLAs.

Definitions of the five parameters are given below:

- SLA Criteria: You can choose any of the five KPI parameters from the SLA Criteria drop-down menu. For example, when you choose Frame Rate, you can select either less than (>) or greater than (<) as an operation and supply a value to complete the calculation. For example, the Frame Rate is > 15.
- Capture FPS: Capture FPS is the number of frames captured per second. Here the value assigned to this parameter is 5 frames per second.
- Sample Time Period: Sample Time Period is the time you assign for the video frame to be captured. Here, the value assigned to this parameter is 10 seconds.

- Capture Time Before: Capture Time Before is the overhead time required before the actual sample time period begins. For example, we have assigned 3 seconds to actually click on the keys, navigate on the device and move from one screen to another, including delay time for uploading of frames, etc. This is before we start the selected video, which would take 10 seconds (the sample time period).
- Capture Time After: Capture Time After is the overhead time period required after the actual sample time period has finished.

6.9.8 Start a Monitor Session

Make sure to check the Enabled checkbox. Then start the monitor by clicking the Start Monitor button under the KPI Parameter Graph.

Start Monitor: This button starts the monitoring, provoking graph generation.

Stop Monitor: This button stops the monitoring (i.e., stops graph generation).

Now interact with the device. Make sure to capture test steps that exercise the KPI Group, the plugin, and the SLAs you've defined. You can start monitoring and viewing KPI parameter values as they are calculated from the video feed. These parameters are represented by different colors that are reflected on the graph above.

As soon as an SLA violation is generated, the Enabled checkbox is unchecked and is reset to checked status.

A log of the SLA violations appears in the SLA Violations window. A SLA Violation specifies the violation rule, the time and date at which it is generated, and the status of the frames. If the KPI parameter values fall outside the SLA range for the moving average defined by the SLA sample period, an SLA violation is generated and it shows up in the SLA Violation list box.

6.9.9 Viewing an SLA Violation

There are two ways to open an SLA violation:

- Right click on the violation to display the drop-down menu and select Open Violation.
- Double click on the violation, and the violation appears in the Video Player.

The violation in the Video Player shows the video data captured during the violation time period, as well as the KPI parameter values that were calculated.

Figure 6-28 Open Violation in Video Player

6.9.10 Composite Monitor Views

When you add several devices to a video monitor view, you can configure and start the monitors in the device view and observe multiple monitors in the composite view.

This layout can be useful for configuring SLAs on different devices using the same KPI group or the same video feed to ensure that the algorithms and SLAs are configured properly for all devices.

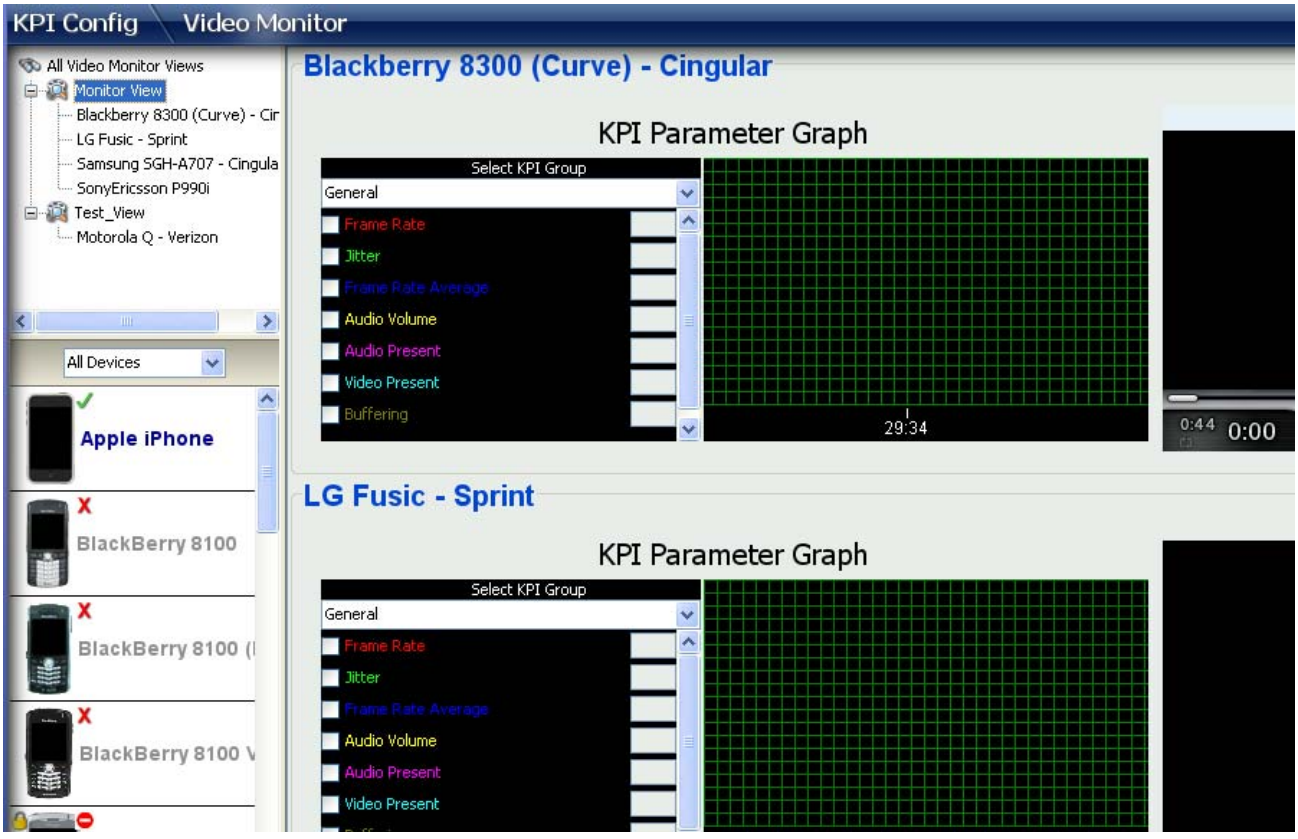


Figure 6-29 Composite View of Video Monitors

6.9.11 VM Commands in Scripts

After you have successfully configured KPI groups and defined useful SLAs, you can use a Production Monitor to leverage VM functionality in a recurring, automated environment. For that, first create a monitoring script to play some video on a device.

6.10 Object-Oriented Scripting

Monitoring scripts enable you to track the performance of your mobile services in real-time. Monitoring scripts are special scripts that measure the time it takes to complete tasks or monitor video performance on a network and in mobile applications.

In addition, monitoring scripts support Video Monitoring (VM). This feature enables you to monitor video performance over the network. For detailed information on Video Monitoring, refer to "Video Monitor" on page 187.

For a full description of script commands, refer to "DeviceAnywhere Script Commands" on page 223.

You can write monitoring scripts using multiple techniques that are supported by *DeviceAnywhere Monitoring*.

- One option is to use the icon-based scripting language. You use the visual script editor to drag and drop script commands into the script.
- Another option is to use the Record and Playback functionality to record scripts by interacting with the device. As you press keys, script commands are automatically added to the script.

The best practice is to use a combination of both of these techniques. Use the Record and Playback toolbar in the script editor.

6.10.1 Adding Script Commands to the Script

All available script commands are located along the left hand side of the script editor.

You can add script commands to the script by dragging them from the left into the script editor.

Figure 6-30 shows a SendKeys command that has been dragged into the script editor.

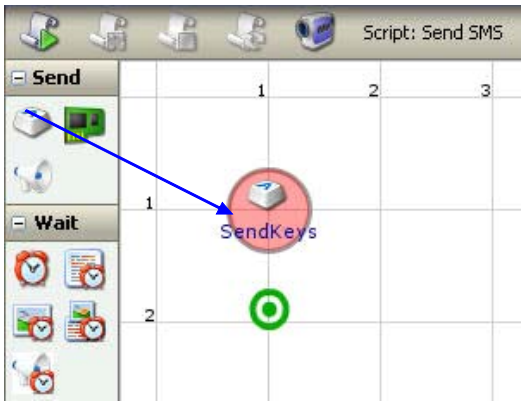


Figure 6-30 SendKeys Command Dragged to the Target

6.10.2 SendKeys Command Example

The SendKeys command can be used to execute key presses on a device. The SendKeys command will be used as an example to illustrate the properties and settings associated to each script command.

First look at the Text To Send field by opening the SendKeys command:

1. Right click on the SendKeys command. The Script Command menu appears.
2. Select Properties. Notice the Text To Send field is empty. This field stores the key presses that are made on the device or text that is manually entered to execute on the device.

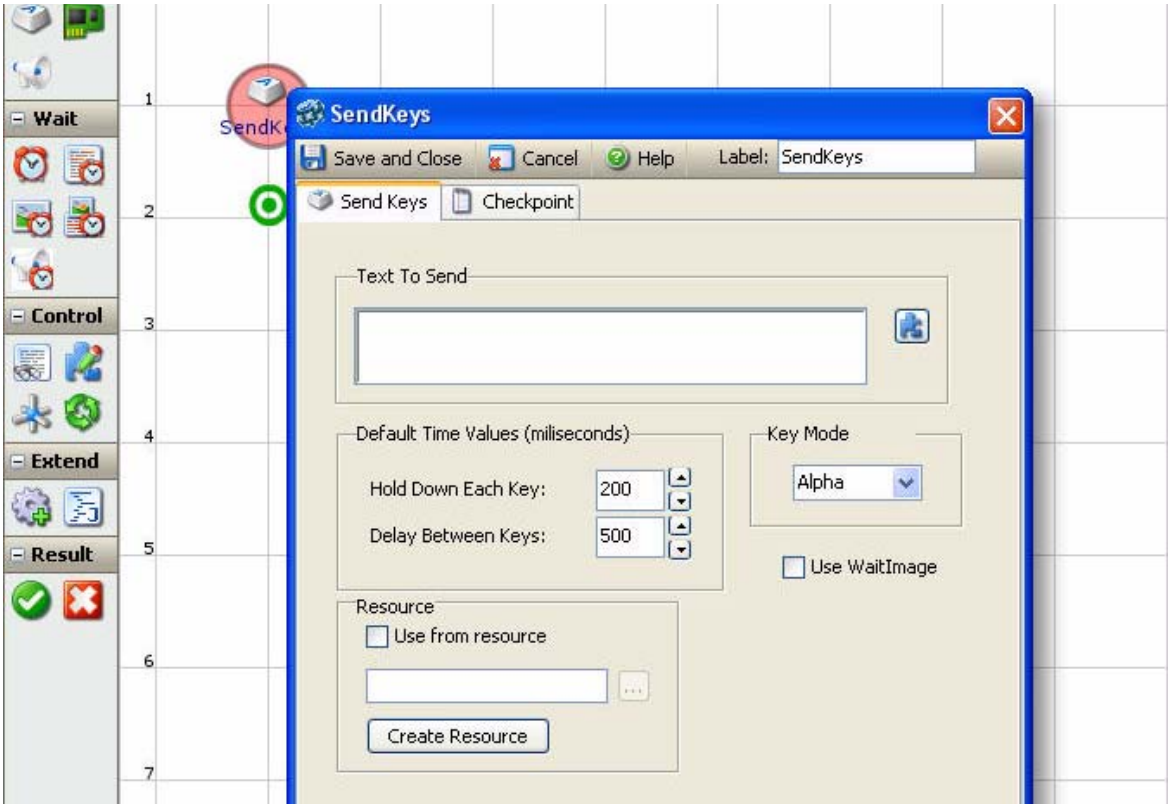


Figure 6-31 Empty Text To Send Field

There are two ways to add key presses to the Text To Send field.

- You can use recording capabilities.
- You can add text by entering it directly into the Text To Send field.

6.10.3 Recording the Key Presses

How to Record Key Presses and Store them in the SendKeys Command:

1. Right click on the SendKeys command. The Script Command menu appears.
2. Select Start Recording from the drop-down menu.
3. Press the Call End key four times on a device added to the script and acquired.
4. Right click on the SendKeys command.
5. Select Stop Recording.

Now take a look at the Properties of the SendKeys command. Double click directly on the Send-Keys command to open the Properties screen. The Text To Send field now contains the four Call End key presses you just recorded.

Note: By right clicking on an individual SendKeys command and starting to record, you force all recorded key presses to be added to a particular SendKeys command. This action is different than starting your recording from the toolbar, where the system places key presses in separate SendKeys commands when there is too much hesitation between key presses. Recording into a single SendKeys command gives you more control over the organization of your test script.

6.10.4 Entering Text Manually in Text to Send Field

In this section, you copy a phone number from the Properties screen of a device, then paste it into the SendKeys Text to Send field.

How to Copy the Phone Number of a Device:

1. Right click on an active device in the workspace. The Properties screen is displayed.
2. Highlight the phone number from the phone number field.
3. Use the keyboard shortcut Control+C to copy the phone number to the Clip board.

Now paste the phone number into the Text to Send field in a new SendKeys command.

How to type or paste text directly into the Text To Send field:

1. Drag a SendKeys command to the test script. Instead of recording your key presses, you are going to manually enter the phone number of the device.
2. Double click on the SendKeys command. The Properties screen is displayed.
3. Paste the phone number that was copied from the device into the Text to Send field. To paste the phone number, press Control+V. After pasting a phone number, remove any hyphens or spaces from the Text To Send field.
4. Make sure that the Key Mode is set to Numeric, since a number is entered in the Text to Send field.
5. Click Save and Close.

6.10.5 Labeling Test Steps

All the labels on the script commands can be changed to make it easy to identify the actions in the test script.

How to Label Test Steps in a Script:

To change the label of a script command, double click the command to open the Properties screen. In the Label field (top right), enter a name that describes the test step. Click Save and Close.

6.10.6 Running a Script with Run from Here

The Run from Here command is an essential step in script building. *DeviceAnywhere Monitoring* allows you to execute a command at any point of a test script.

Use the Run from Here command in two ways:

- When you have typed or pasted text into a SendKeys, you must have the device execute the key strokes to get the device into the correct state. You have to set the device correctly before you can go on to the next step in the test script.
- When you are debugging, Run from Here is a great way to ensure that a long series of key presses runs correctly.

How to Select Run from Here:

1. Right click on the script command. The Script Command menu appears.
2. Select Run from Here.

The key presses entered into the Text To Send field are executed on the device.

6.11 Script Editor

The Script editor contains two toolbars and a workspace where you edit.

- The Script Command Toolbar, on the immediate left of the Script editor, contains the script commands that allow you to build a script.
- The Record and Playback Toolbar, above the Script editor, contains the controls to start or stop recording and to play or stop the execution of a script.
- The workspace is shown with a grid in which the script commands are placed. You can adjust the grid size using Zoom in the Menu Bar and selecting the zoom percentage.

6.11.1 Record and Playback Controls

Figure 6-32 illustrates the Play, Stop, Pause, and Restart buttons in the toolbar that control script execution.

Figure 6-32 Play, Stop, Pause, and Restart Buttons

The Record and Playback toolbar has five controls (from left to right):

- Play Script—This control executes a script. Play Script is disabled when a script is already executing.
- Pause Script—This control enables pause of an execution of a script. Pause Script is only enabled during script execution.
- Stop Script—This control stops the execution of a script. Stop Script is only enabled during script execution.
- Restart Script—This control allows you to restart the execution from the start. Re-Run Script is available during script execution.
- Record Script— Click on Record Script, and a recording light blinks on the button indicating that recording has begun.

6.12 Working with Scripts

While working on a script, you may need to save changes, discard changes to start over, or export the script to your computer. The Script menu provides these options.

6.12.1 Script Status

Scripts are available for editing or saved in the database. A yellow lock on the script icon indicates that the script is open for editing. No other user has access to the script in the database. After you save a script, the yellow lock is removed. Refer to “How to Open a Script for Editing” below.

6.12.2 Open the Script Menu

How to Open the Script Menu:

To open the Script menu, right click on a script in the Explorer.

The Script menu has the following options.

- Rename Script—Rename Script lets you modify the name of a script.
- Save Changes—Save Changes saves the script to the database.
- Copy Script—Copy script in the editor.
- Cut Script—Cut script from the script editor.
- Paste Script—Paste script into the script editor.
- Export Script—The Export Scrip option allows you to export the script to your computer.
- Discard Changes—Discard Changes lets you get rid of all changes in the current script.
- Delete Script—Delete Script removes the script from the Explorer.
- Properties—This option opens the Script Properties screen. This screen provides a number of settings to view the history of scripts in a test environment. It also allows you to export and import data sets to be used in scripts. Refer to “Script File Properties” on page 201 for further details.

6.12.3 Perform Simple Tasks with the Script Menu

The following procedures describe how to use the Script menu to perform simple tasks.

Rename a Script

How to Rename a Monitor Script:

1. Right click on the script. The Script menu appears.
2. Select Rename Script. The file name is open for editing.

Save a Script

How to Save a Monitor Script to the Database:

1. Right click in the script editor. The Script menu appears.
2. Select Save Changes. The yellow lock is removed from the script icon in the Explorer.

Open a Script for Editing

How to Open a Monitor Script for Editing:

1. Right click on the script in the Explorer.
2. Select Lock/Edit Script. The yellow lock appears on the script icon in the Explorer.

Export a Script

How to Export a Monitor Script:

1. Right click on the script in the Script Editor.
2. Select Export. A Save file entry screen is displayed.
3. Navigate to the folder where you want the script saved.
4. Enter the name of the script.
5. Click Save.

The monitor script is saved to your computer.

6.12.4 Monitor Script Properties Screen

The Monitor Script Properties screen contains a number of settings to control and review the history of scripts in a test environment. These tabs allow you to add comments about a script. Four tabs are available in this view.

How to Open the Monitor Script Properties Screen:

1. Right click on a script in the Explorer. The Script menu appears.
2. Select Properties.

Figure 6-33 illustrates the Monitor Script Properties screen.

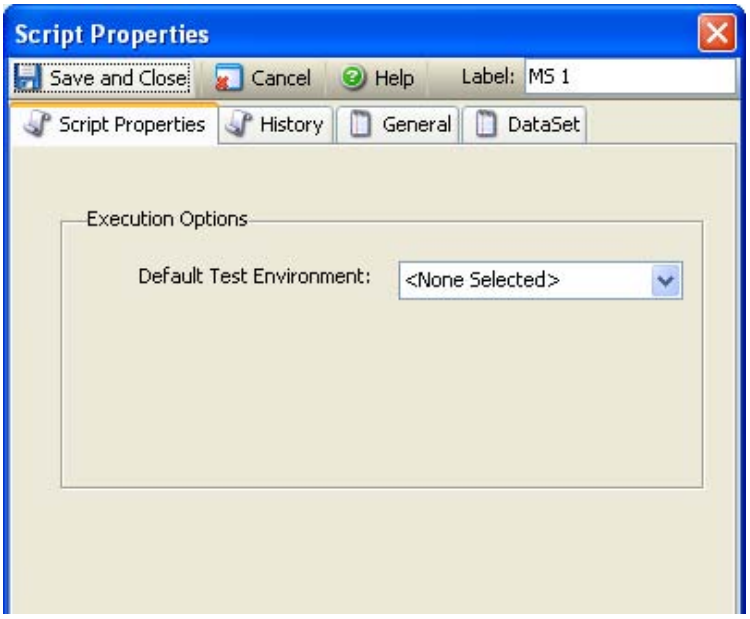


Figure 6-33 Monitor Script Properties Screen

6.12.5 Properties Screen Toolbar

The Properties screen toolbar at the top supports basic commands to save changes, cancel changes, and access online Help while working in the Properties screen. A brief description of these commands follows:

- Save and Close—This command lets you save the settings you entered into the Properties screen.
- Cancel—This command allows you to dismiss the Properties screen without saving any of the changes you made.
- Help—This command provides access to online Help.
- Label—The Label entry field lets you enter the name for each SendKeys.This is a useful feature to label the steps in a long script.

All Properties screens in *DeviceAnywhere Studio* contain this toolbar and function in the same way.

6.12.6 Default Test Environment

DeviceAnywhere Studio supports scripting efforts for three test environments:

- Production
- Staging
- Development

When you first create a script, you may want to set the test environment that the script is written for. The test environments represent the stages of development. Test environment is used like a global parameter. When the test environment is moved from one stage to another, the scripts can be promoted or validated for a new test environment.

6.12.7 Script History

The History tab provides access to all versions of the script stored in the database. Reviewing the history of a script is a good way to refer to a previous working copy of a script, in case subsequent changes break its functionality.

From the History tab you can checkout a previous version or delete a version of a script. Once the database is changed, these changes become permanent.

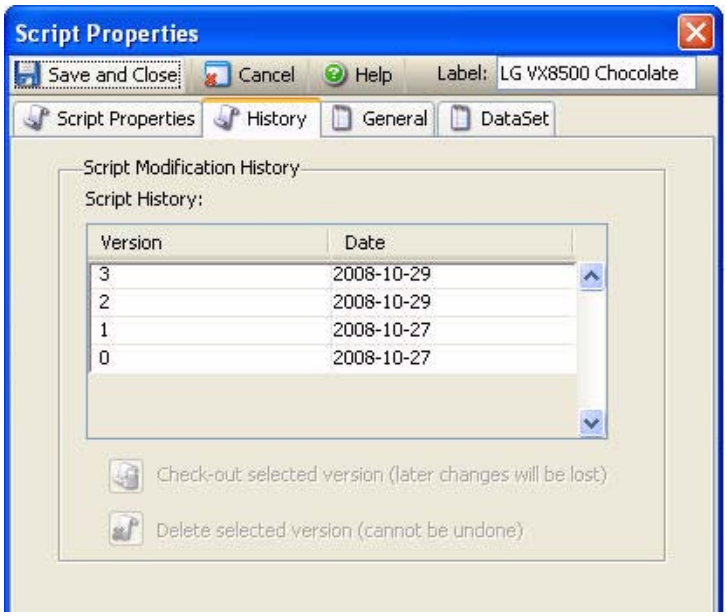


Figure 6-34 History Tab in Script Properties

6.12.8 Script File Properties

The General tab shows file properties such as creation date and the last date modified for a script. This tab also has an entry field to enter comments.

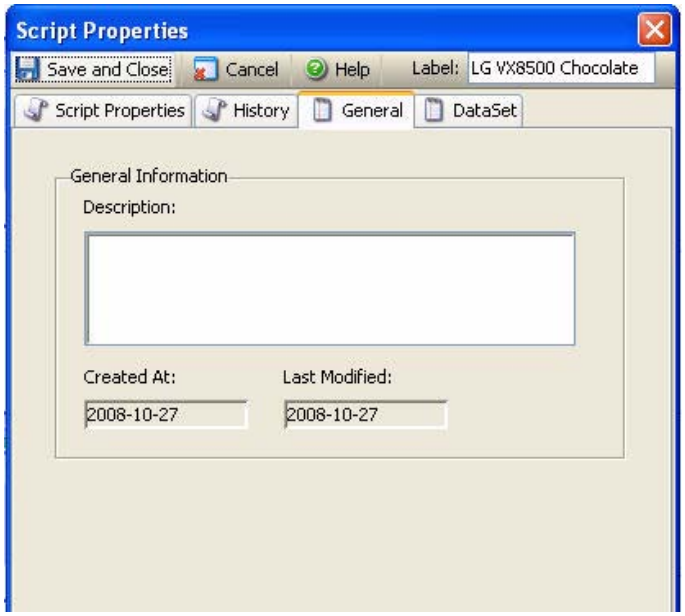


Figure 6-35 General Tab in Script Properties

6.12.9 Dataset

This tab is used to import data for data-driven scripts.

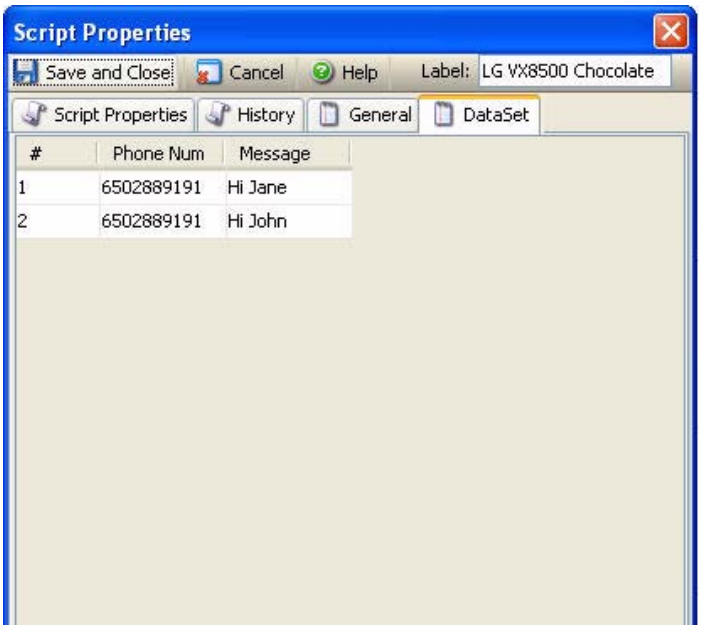


Figure 6-36 Dataset Tab in Script Properties

6.13 Send Message Script with Timers

Figure 6-37 shows the steps in the Send Message script. It is a visual representation of the steps needed to send a text message. It also shows Toggle Timer commands inserted before sending an SMS message and after receiving a message alert. The Start/Stop Timer value is compared to the SMS Timing SLA violation rules to measure the performance of the Send Message script.

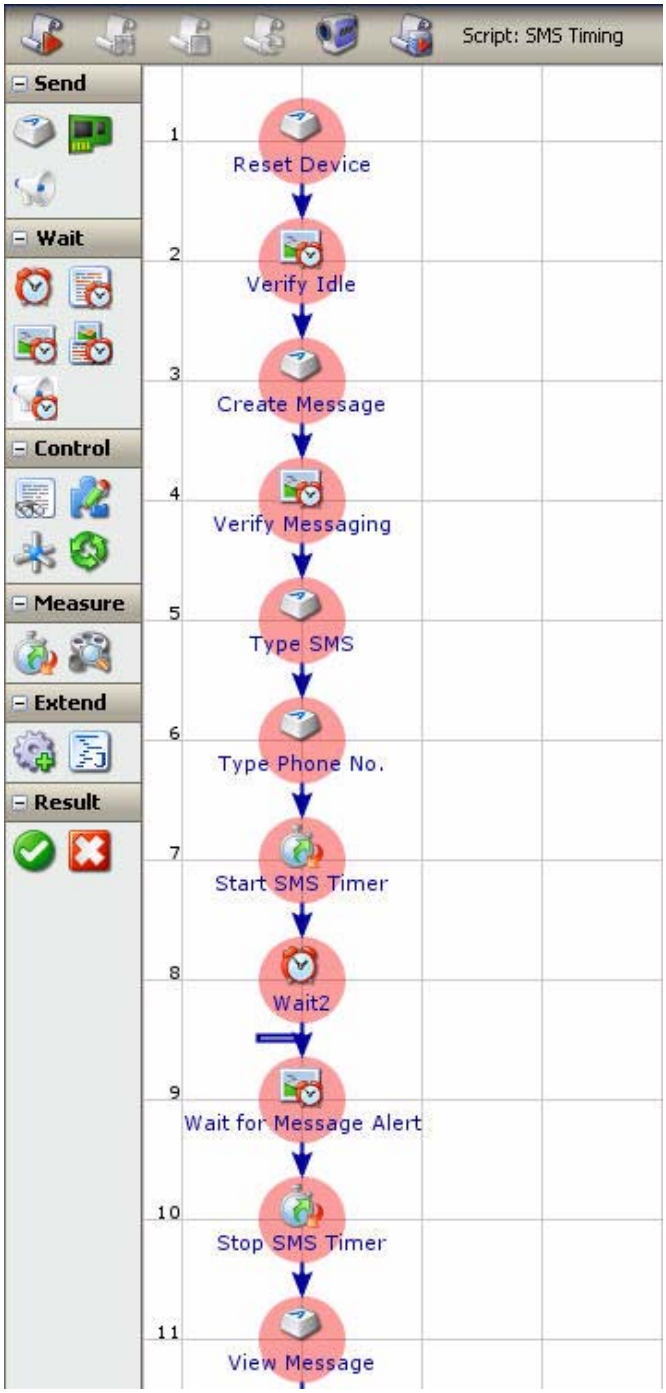


Figure 6-37 Send Message Script with Toggle Timers

Table 6-1 provides a written description of the steps to send a text message.

Table 6-1 Send Message Script with Toggle Timers

Action	Step	Expected Results
1. Reset Device	Drag SendKeys and record the End key four times	Phone returns to idle screen
2. Verify Idle	Drag WaitImage and capture reference image for compare	Reference image for idle screen saved.
3. Create Message	Drag SendKeys and record key presses to open menu and select Messaging screen	Key presses stored to access the messaging screen.
4. Verify Messaging	Drag WaitImage and capture reference image for compare	Reference image for messaging screen saved.
5. Type SMS	Drag SendKeys and enter text message: Automated SMS test	Enter text message to be sent to device.
	Run from Here	The message is entered into the device.
	Right click on the	Send Automated SMS test to the device.
6. Type Phone Number	Copy phone number of the Samsung A707	Copy phone number to send text message to.
	Drag SendKeys and paste phone number into the Text to Send field	Prepare to send phone number to the Motorola L6.
	Run from Here	Sends the phone number to the device.
	Record key presses to tab from the phone number field on the phone to the text message entry screen.	Prepare to enter text message.
7. Start SMS Timer	Drag a Toggle Timer and configure to a “one time” start timer.	Clock starts counting for key presses executed.
8. Wait2	Drag a Wait command into the script to synchronize with the Message Alert screen.	Scripts waits for a specified time.
9. Wait for Message Alert	Drag WaitImage and capture reference image for compare.	Reference image for Message Alert screen saved.
10. Stop SMS Timer	Drag a Toggle Timer and configure to a “one time” stop timer.	Clock stops time and retains elapsed time value to send an SMS message.

Figure 6-38 Toggle Timer Properties Screen

Figure 6-38 shows the Toggle Timer Start parameter selected from the drop-down menu. This action configures the Start SMS Timer.

How to Configure the Start SMS Timer:

1. Double-click on the Start SMS Timer to open the Properties screen.
2. Click on the Timers tab.
3. Enter the name of the Timer. In this case, Start SMS Timer.
4. Check the “one time” indicator.
5. Click the Toggle Timer tab.
6. From the drop-down menu, select the Start Timer.
7. Click Save and Close.

How to Configure the Stop SMS Timer:

1. Double-click on the Stop SMS Timer to open the Properties screen.
2. Click on the Timers tab.
3. Enter the name of the Timer. In this case, Stop SMS Timer.
4. Check the “one time” indicator.
5. Click the Toggle Timer tab.
6. From the drop-down menu, select the Stop Timer.
7. Click Save and Close.

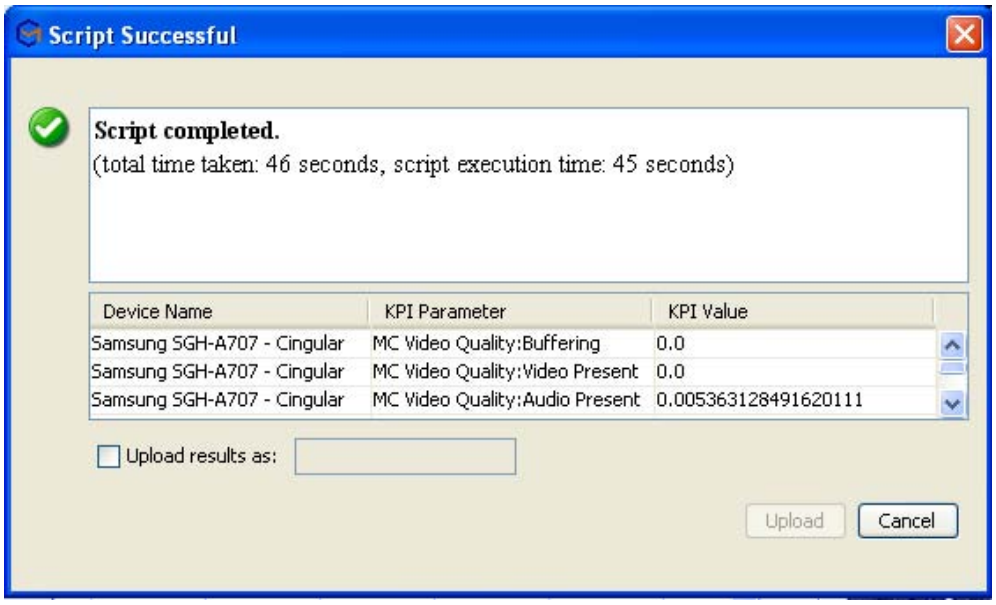


Figure 6-39 Monitor Script Results

To upload the test results for further analysis, do the following:

1. Click the checkbox before Upload results as.
2. Enter a name for your test results in the field Upload results as.
3. Click Upload.

After clicking Upload, a prompt screen is displayed. Click View Now and you are automatically taken to the *MyDeviceAnywhere* web site where you can view the results. Test results are shown under the Test Results tab in your *MyDeviceAnywhere* account.

Click Cancel when you want to dismiss the prompt screen and remain in *DeviceAnywhere Monitoring*.

Refer to “Scheduler” on page 159 for detailed information on creating a Production Monitor.

6.14 Building an ESPN News Monitoring Script

In this section, you develop a complex monitoring script to analyze an ESPN News video and to measure critical timing for downloading and buffering the video.

When you are done, the monitoring script will include many SendKeys, WaitImages, Toggle Timers and VM pair commands in sequence. The script commands have been relabeled to more easily identify the steps in the script. The ESPN News script also includes the NavigateTo command.

In this example, the Samsung A707 device was used. However, you can follow this script in a general way. Each device has a unique series of key presses to access News features. Therefore, take into consideration the device you are using and create your script accordingly.

Figure 6-40 shows the top portion of the ESPN News monitoring script. It is a visual representation of steps needed to access the ESPN video menu and download an ESPN Sports video clip. The script has 40 steps in all. This chapter presents only parts of the ESPN News script and focuses on the use of the Toggle Timer commands specifically.

Table 6-2 below provides a detailed description of the steps in the ESPN News monitoring script. This table provides an overview of every step needed to develop a successful monitoring script, including Wait script commands that are programmed to wait a few seconds and inserted to adjust for network response time.

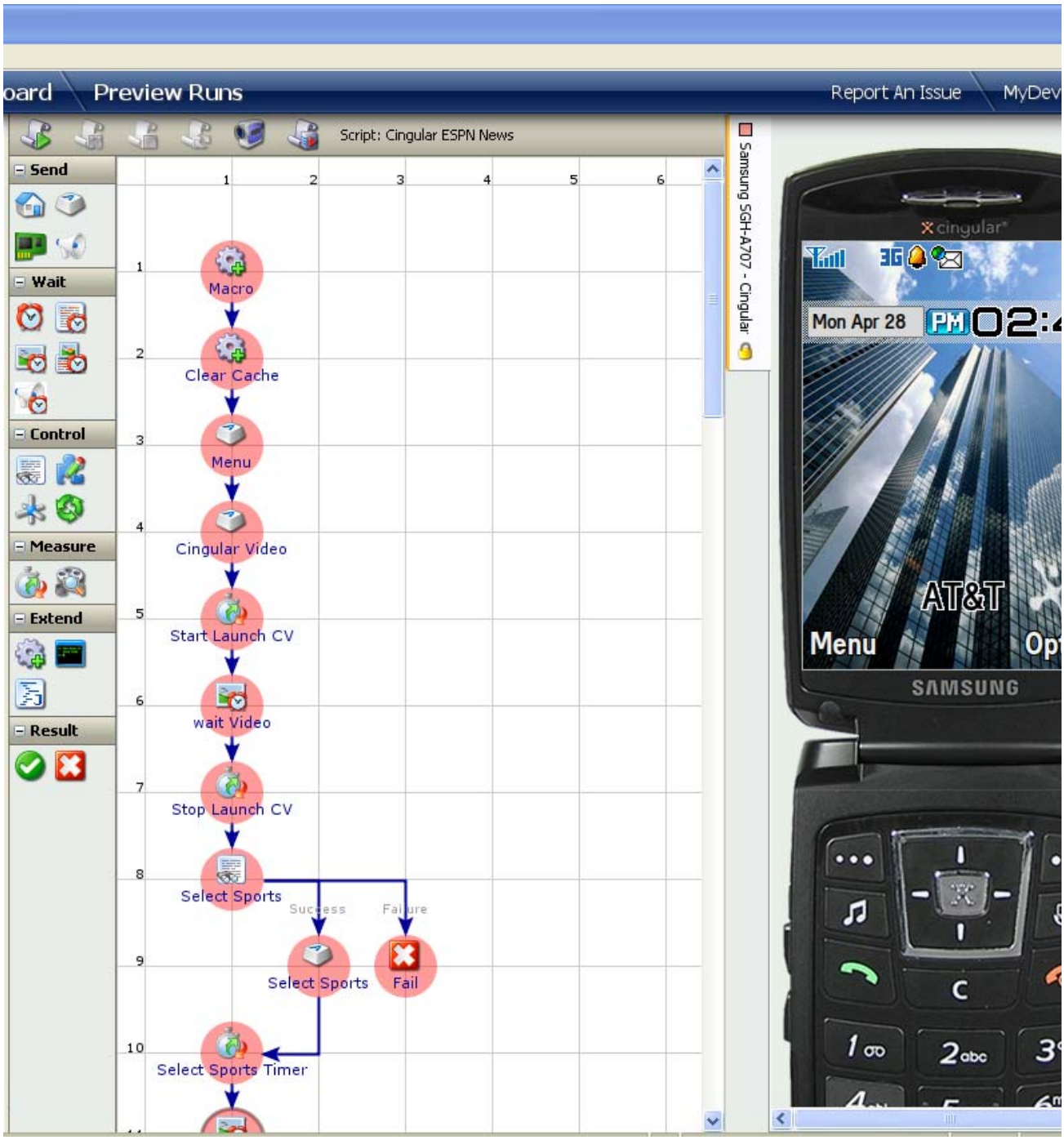


Figure 6-40 ESPN News Script

Table 6-2 provides a written description of the steps in the ESPN News monitoring script.

Table 6-2 ESPN News Monitoring Script

Action	Step	Expected Results
1. Macro	Reset device by storing End key presses.	Device returns to idle screen
2. Clear cache	Access the MediaNet Advanced settings to clear the cache. Select clear cache and confirm. The confirmation screen appears on the device screen.	Device executes key presses to select cache, delete cache, and return the confirmation screen.
3. Menu	Return to main menu.	Device accesses the main menu.
4. Cingular Video (CV)	Select the Video service.	Device navigates to Cingular Video and selects CV.
5. Start Launch CV (Toggle Timer)	Start Toggle Timer inserted to start the time taken to open the CV.	Toggle Timer starts clock. This Toggle Timer is matched with step 7.
6. Wait Video	WaitImage inserted to compare the device screen with the reference image of the CV main menu.	Reference image compared with CV main menu and goes to next step when successful.
7. Stop Launch CV (Toggle Timer)	Stop Toggle Timer inserted to stop timing on the time taken to open the CV.	Toggle Timer stops the clock. This Toggle Timer is matched with step 5.
8. Select Sports (NavigateTo)	Navigate to the Sports video offerings. The NavigateTo command executes [select] in the SendKeys tab and compares the reference image of Sports option in the CV video menu.	Device navigates to Sports and selects the Sports option.
9a. Success—Select Sports	Access Sports option	Device selects the Sports option.
9b. Failure—Generate fail message	WaitImage fails to match the reference image for the Sports main menu. The script exits with a fail message.	Device fails to access the Sports main menu.
10. Select Sports Timer (Toggle Timer)	Start Toggle Timer inserted to start the time taken to open the Sports option.	Toggle Timer starts the clock. This Toggle Timer is matched with step 12.
11. Wait for Sports	WaitImage inserted to compare the device screen with the reference image of the Sports main menu.	Reference image compared with Sports main menu and goes to next step when successful.
12. Stop Sports Timer (Toggle Timer)	Stop Toggle Timer inserted to stop the time taken to open the Sports option.	Toggle Timer stops the clock. This Toggle Timer is matched with step 10.

Table 6-2 ESPN News Monitoring Script (Continued)

Action	Step	Expected Results
13. NavTo ESPN	Navigate to the ESPN channels. The NavigateTo command executes [select] in the SendKeys tab and compares the reference image of ESPN channels menu with the device screen.	Device navigates to the ESPN channels.
14a.Success—Select ESPN	ESPN channels selected.	Device screen shows ESPN channels menu.
14b. Failure—Generate fail message	WaitImage fails to match reference image for ESPN channels menu. The script exits with a fail message.	Device fails to access the ESPN channels menu.
15. Wait for ESPN	WaitImage inserted to compare the device screen with the reference image of the ESPN channels menu.	Reference image compared with ESPN channels menu and goes to next step when successful.
16. Wait3	Wait inserted to account for network response time. This Wait set to 30 seconds.	Script waits for 30 seconds.
17. NavTo ESPN News	Navigate to the ESPN News channel. The NavigateTo command executes [select] in the SendKeys tab and compares the reference image of ESPN News option with the device screen.	Device navigates to the ESPN News channel.
18a. Success—Select ESPN News.	ESPN News selected.	Device screen shows ESPN News video clips.
18b. Failure—Generate fail message	WaitImage fails to match reference image for ESPN News video clips. The script exits with a fail message.	Device fails to access the ESPN News video clips.
19. Wait for ESPN News	WaitImage inserted to compare the device screen with the reference image of the ESPN News video clips.	Reference image compared with ESPN News video clips and goes to next step when successful.
20. Wait2	Wait inserted to account for network response time. This Wait set to 3 seconds.	Script waits for 3 seconds.
21. Select Video clip	SendKeys executes down arrow and select for the first video clip.	The first video clip is selected.
22. Start Play Video (Toggle Timer)	Start Toggle Timer inserted to start the time taken to download the video clip.	Toggle Timer starts clock. This Toggle Timer is matched with step 28.

Table 6-2 ESPN News Monitoring Script (Continued)

Action	Step	Expected Results
23. Connected	WaitImage inserted to compare the device screen with the reference image of video clip download.	Reference image compared with video download screen and goes to next step when successful.
24. Connecting start (Toggle Timer)	Start Toggle Timer inserted to start the time taken to buffer video clip.	Toggle Timer starts the clock. This Toggle Timer is matched with step 27.
25. Buffering	WaitImage inserted to compare the device screen with the reference image of the buffering screen.	Reference image compared with buffering screen and goes to next step when successful.
26. Connecting finish	WaitImage inserted to compare the device screen with the reference image of the end buffering screen.	Reference image compared with end buffering screen and goes to next step when successful.
27. Connection finish (Toggle Timer)	Stop Toggle Timer inserted to stop the time taken to buffer video clip.	Toggle Timer stops the clock. This Toggle Timer is matched with step 24.
28. Stop Play Video	Stop Toggle Timer inserted to stop the time taken to download the video clip and view it.	Toggle Timer stops clock. This Toggle Timer is matched with step 22.
29. VM Start	VM command is inserted to start monitoring video clip.	VM monitoring starts analyzing video clip. This VM command is matched with step 32.
30. Played for 2 seconds	WaitImage inserted to compare the device screen with the reference image of the ESPN video clip News header.	Reference image compared with ESPN video clip News header and goes to next step when successful.
31. 60 seconds	Wait command is inserted to view video clip for 60 seconds.	Device screen shows video clip running for 60 seconds.
32. VM Stop	VM command is inserted to stop monitoring video clip.	VM monitoring stops analyzing video clip. This VM command is matched with step 29. Video and audio is saved to AVI format.
33. Option2	Open the Option menu in video clips.	Option menu is selected in video clips.
34. Wait	Wait inserted to account for network response time. This Wait set to 1 seconds.	Script waits for 1 seconds.
35. Stop3	Select Stop (Option 2) in video clips.	Device stops running video clip.
36. Start Exit Video (Toggle Timer)	Start Toggle Timer inserted to start the time taken to exit video clip menu.	Toggle Timer starts the clock. This Toggle Timer is matched with step 39.

Table 6-2 ESPN News Monitoring Script (Continued)

Action	Step	Expected Results
37. Wait Video Main	WaitImage inserted to compare the device screen with the reference image of the CV main menu.	Reference image compared with CV main menu and goes to next step when successful.
38. Exit Video	Select Option menu and press End key.	Device returns to idle screen.
39. Stop Exit Video (Toggle Timer)	Stop Toggle Timer inserted to exit video clips menu and return to the idle screen.	Toggle Timer stops clock. This Toggle Timer is matched with step 36.
40. Home	Reset the device by pressing the End key 3 times.	Device returns to the idle screen.

It is important to return the device to an idle state where your next test script can start from. As a general rule, add the steps at both the beginning and the end of the script to return the device to a known state.

It is necessary to configure the matching pair of Toggle Timers or VM commands with the same parameters, otherwise unpredictable results may occur.

The procedures below describe how to get started on an ESPN News monitoring script. They describe a portions of the script in detail specifically with Toggle Timers. The other script commands are configured based on core scripting functionality in *DeviceAnywhere Studio*.

Refer to "Automation" on page 77 for additional examples of scripting.

6.14.1 Create New Script and Acquire a Device

How to Create a New Script and Acquire a Device:

1. Right click on a folder.
2. Select New Script
3. Enter the name of the new script.
4. Right click in the Device list.
5. Select the Samsung A707.
6. Select Add to Script.
7. Right click on the Samsung A707 to display the Device menu.
8. Select Access Device.
9. Select Acquire Device.

6.14.2 Clear Macro

The first step in this script is a Clear Macro. The Macro was previously defined in the Macro library. Its function is to clear the device to an idle state by executing [End] key presses on the Samsung A707.

How To Use the Clear Macro in the Monitoring Script:

1. Drag a Call Macro command into the script editor.
2. Double click on the Call Macro to open the Properties screen.
3. Use the drop-down menu to select Clear Macro in the Macro field.
4. Change the label of the Call Macro command by entering Clear Macro in the Label field.
5. Click Save and Close. The Call Macro command is configured to call the Clear Macro script from the Macro library.

In the Call Macro Properties screen, only the name of the Macro is visible. Go to the Macro library and open the Macro to determine the key presses it contains.

In this case, the Clear Macro contains [END][END][END] key presses to return the device to the idle state.

6.14.3 Clear Cache

The next step is to clear the cache of the Samsung A707. This series of key presses is device dependent.

How To Use the Clear Cache Macro in the Monitoring Script:

1. Drag a Call Macro command into the script editor.
2. Double click on the Call Macro to open the Properties screen.
3. In the Macro field, use the drop-down menu to select Clear Cache.
4. Change the label of the Call Macro command by entering Clear Cache in the Label field.
5. Click Save and Close. The Call Macro command is configured to call the Clear Cache script from the Macro library.

To create the Clear Cache Macro for the Samsung A707, record the following key presses. The commands in brackets correspond to command syntax entered in the Text to Send field.

Table 6-3 Clear Cache Macro

Action	Step	Expected Results	Motorola L6
1. Init	Press the End key three times.	Device returns to idle screen.	[END][END][END]
2. Main Menu	Drag a SendKeys command and enter key presses to open the main menu.	The device main menu is displayed.	[Select]

Table 6-3 Clear Cache Macro (Continued)

3. Verify Main	Drag WaitImage and capture reference image for compare to the main menu.	Reference image for main menu screen saved.	WaitImage command succeeds.
4. GoTo Browser Menu	Drag a SendKeys command and enter key presses to navigate to the WebAccess icon.	The WebAccess menu is opened.	[Right][Select]
5. Verify Browser Menu	Drag WaitImage and capture reference image to compare to the WebAccess menu.	Reference image for the WebAccess menu screen saved.	WaitImage command succeeds.
6. Select Browser Setup	Drag a SendKeys command and enter key presses to select Browser Setup in the WebAccess menu.	The device navigates to the Browser Setup option.	[Down][Down] [Down][Down] [Select]
7. Verify Bowser Setup	Drag a WaitImage and capture a reference image to compare to the Browser Setup menu.	Reference image for the Browser Setup menu is saved.	WaitImage command succeeds.
8. Reset Cache	Drag a SendKeys command and enter key presses to navigate to Reset cache and execute Reset cache.	The Reset Cache option is selected and the cache is cleared.	[Down][Down] [Select]
9. Reset Device	Drag a SendKeys command and enter key presses to return the device to the idle state.	The device returns to the idle screen.	[SoftL][SoftL] [SoftL]

Figure 6-41 shows the Samsung A707 indicating the cache is cleared.

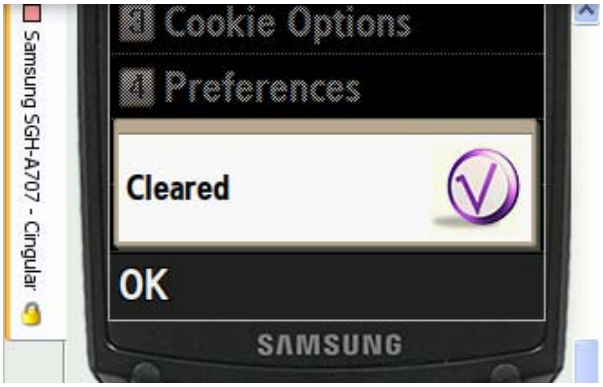


Figure 6-41 Samsung A707 Cache Cleared

6.14.4 Menu

Next, open the main menu of the Samsung A707. This step requires a SendKeys command.

To record key presses to access the main menu, do the following:

1. Drag a SendKeys command to the green target.
2. Right click on the SendKeys command.
3. Click Start Recording from the drop-down menu.
4. On the Samsung A707, press Open menu.
5. Click Stop Recording in the toolbar.

6.14.5 Cingular Video

1. Drag another SendKeys command into the editor.
2. Right click on the SendKeys command.
3. Click Start Recording from the drop-down menu.
4. Now press the arrow key to navigate to CV in the main menu.
5. Select Cingular Video.
6. Click Stop Recording.
7. To change the label of the SendKeys, enter Cingular Video in the Label field.
8. Click Save and Close.

6.14.6 Start Launch CV

This time the script calls for a Toggle Timer to measure the time required to open the Cingular Video menu on the device.

1. Drag a Toggle Timer command into the script editor.
2. Double click the Toggle Timer to open the Properties screen.
3. Set the Toggle Timer to a Start timer.
4. Click on the Timers tab.
5. Select Add new timer.
6. Enter the name of the timer in the entry field.
7. Select the timer type: One time, Average, or Summation.
8. Change the label of the Toggle Timer command by entering Start Launch CV in the label field.
9. Click Save and Close.

Figure 6-42 shows the Toggle Timer configured as a Start Timer.

Figure 6-42 Start Launch CV Toggle Timer

Figure 6-43 illustrates the Timer tab in the Start Launch CV Properties screen.

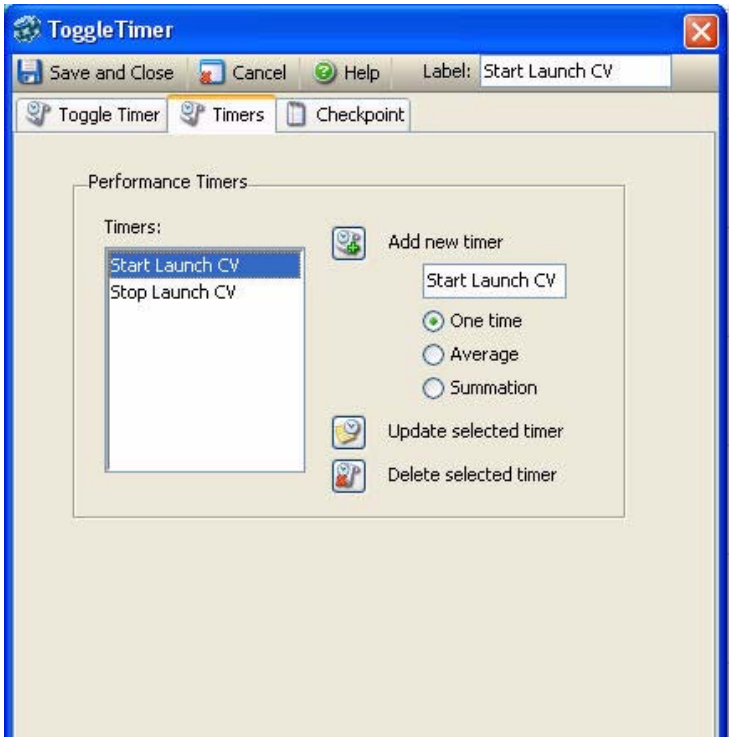


Figure 6-43 Toggle Timers

6.14.7 Wait Video

Wait Video is a WaitImage script command, which serves as a checkpoint to determine whether the monitoring script accessed the Video services main menu.

The WaitImage Properties screen has a number of settings to configure the WaitImage. It allows you to select an area of the reference image that will be used in the compare function. It also allows you to indicate where *DeviceAnywhere* scans for the compare function.

To open the WaitImage Properties screen, double click on the WaitImage. The WaitImage Properties screen appears.

Figure 6-44 Wait Video Properties Screen

The following settings are available on the WaitImage Properties screen:

- Selection Controls—This button is used to select a region of the reference image for the compare function. Click Selection Controls, then draw a rectangle on the image on the left to indicate the region that will be used in the compare function. To discard the region, click the Selection Controls button again.
- Camera—When the SendKeys is first added, the current device screen is captured. This reference image can be updated to the most current device screen by pressing the Camera button. If a new image is available, it will be displayed. You can pause the live recording, navigate to the correct frame by pressing the Device Video Controls, and return to live recording. This procedure allows you to enter any device screen that was previously recorded in the SendKeys command.
- Import—This option allows you to import an image from your desktop.
- Resource—These commands in the Resource window allow you to browse for an image in the Resource Manager or create a resource by storing the reference image from the SendKeys command.

6.14.8 Set the Reference Image

Comparing a small region of the reference image shortens processing time for the compare function.

How to Set the Reference Image:

1. Double click the WaitImage. The WaitImage Properties screen appears.
2. Click Selection Controls.
3. Draw a rectangle over the image (on the left) with the cursor to indicate the region of the image that will be used in the compare function. This included region is highlighted.
4. Choose the location of the search: Fixed or Any Position (which scans the entire image for a match).
5. Click Save and Close.

The WaitImage is set to a region of the reference image for a compare function.

6.14.9 Stop Launch CV

The Stop Launch CV is configured in the same way as Start Launch CV. It is necessary to configure the matching pair of Toggle Timers with the same parameters, otherwise unpredictable results may occur.

6.14.10 Save Changes to the ESPN News Script

DeviceAnywhere Studio has version control built into the product. Scripts are automatically saved to the database as you are working. However, Saving Changes writes an official copy of the script to the database for the script History. When you write the script to the database, the Lock icon is removed from the script to indicate that it is released for editing. Any team member can access the script at this point.

For scripts, right click the script in the Explorer and select the Lock/Edit command to edit the script further after saving changes.

Now continue building the ESPN News script from the steps listed in Table 6-2.

6.14.11 Play Your New ESPN News Script

Play your script by pressing the Play button in the toolbar. Make sure your script executes from start to finish. If it doesn't complete successfully, take the time to understand why the script fails and make the appropriate modifications. Sometimes you may need to insert more Wait time between other script commands to ensure the script is synchronized with network response times.

Key points:

- The WaitImage command checks to see if the defined reference image appears on the device.
- The reference image can be looked for at a fixed location or any position. Any position is useful when looking for images that may move around the screen (dynamic lists for example).
- The WaitImage command continues to look for the image until the Timeout value is reached.
- It is important to use WaitImage commands along the way to ensure that the script is behaving properly and also to make sure that commands are not executing on the device before it is ready to accept input.
- You don't have to run the script from the beginning during script development. You can run from any intermediate step in the script. This is great for debugging and entering copied text into a device.
- Key Mapping is an important feature within *DeviceAnywhere*. When entering any text into the device as part of a script, you must select the correct mode (Alpha for text, Numeric for numbers, and Web for URL entry). The default Key Mode is Alpha.
- It is important to return the device to a state where your next script will start from. As a general rule, add the steps at the beginning of the script to return the device to a know state and also do it at the end of the script.

6.14.12 Upload Test Results

This section describes how to view the results of a monitoring script that was executed.

- Upload the results to your *MyDeviceAnywhere* account
- View the results with images

The Monitor Script Results allows you to select options to upload test results.

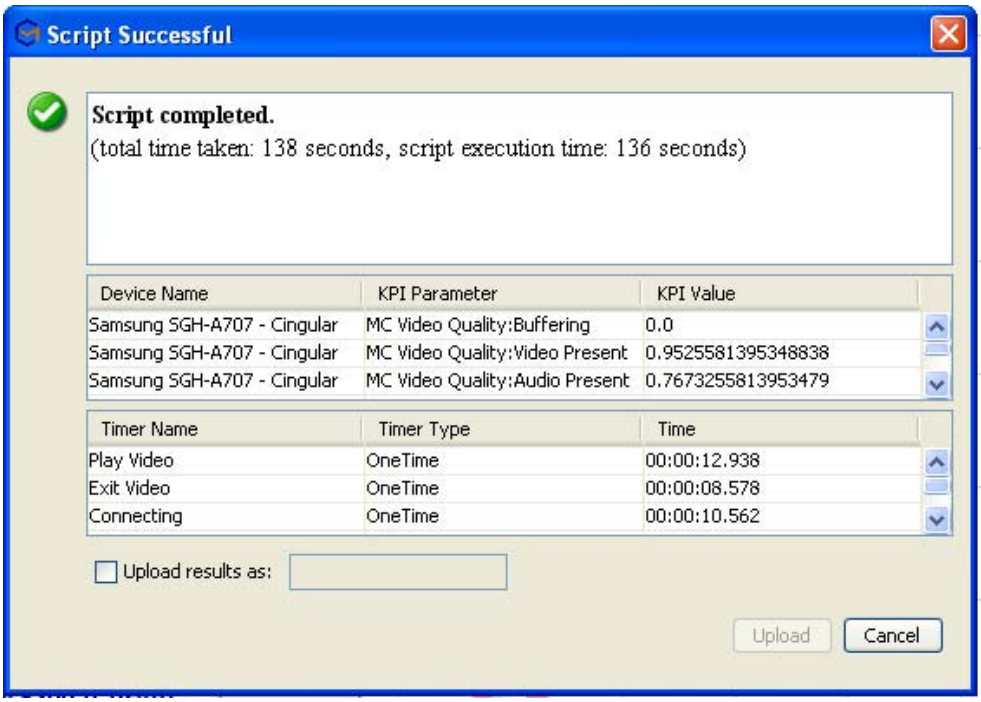


Figure 6-45 Monitor Script Results

To upload the test results for further analysis, do the following:

1. Click the checkbox before Upload results as.
2. Enter a name for your test results in the field Upload results as.
3. Click Upload.

After clicking Upload, a prompt screen is displayed. Click View Now and you are automatically taken to the *MyDeviceAnywhere* web site where you can view the results. Test results are shown under the Test Results tab in your *MyDeviceAnywhere* account.

Click Cancel when you want to dismiss the prompt screen and remain in *DeviceAnywhere Monitoring*.

6.14.13 Viewing Production Monitoring Results

When you run a monitoring script in a Production Monitor, the results are automatically uploaded to the Production Monitoring tab in the *MyDeviceAnywhere* web site

Within the Production Monitoring tab, four links are available in the toolbar:

- View Data
- Run/Schedule Reports
- View Saved Reports
- Execution Status

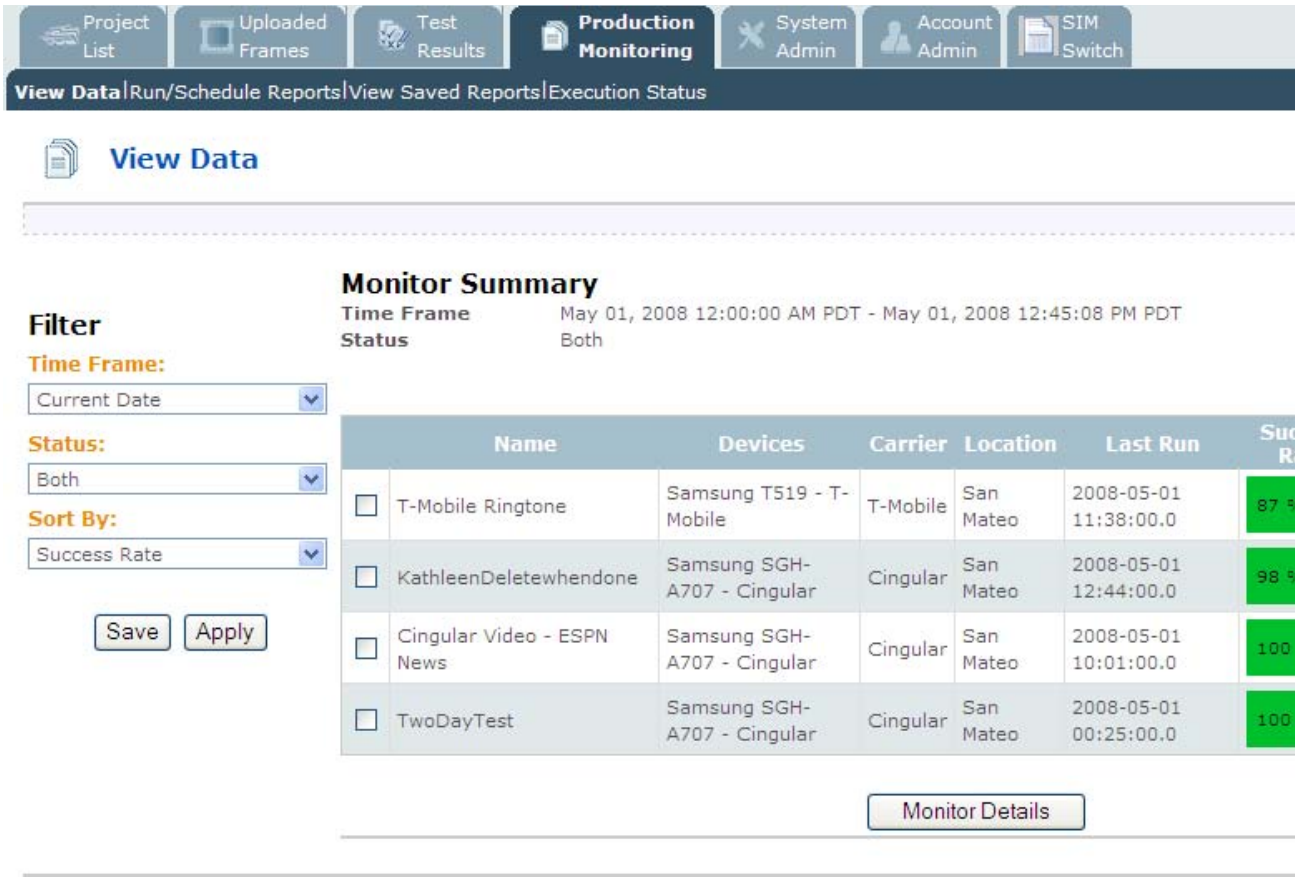


Figure 6-46 Monitor Summary

You can click on a monitor to view a more detailed report.

The result details view (not shown) includes the following information:

- Script Info—This displays the name of the script, the device make and model, and the carrier.
- Run Info—This displays the summary of the result; project name, test environment, outcome, run by, start time, duration and description.
- Step-By-Step—This section provides step-by-step results of the entire script. It lists Data, Start, End Time and Images against each step to give you a clear picture of what happened during the test. Clicking on View will show the resulting images for the step.
- Follow-Up History—Comments posted by different viewers on the test are available here.
- Follow-Up Comments—You can add your comments to the various runs here, and save them by clicking the Save button on the page.

7 DeviceAnywhere Script Commands

7.1 Script Commands Overview

DeviceAnywhere supports both simple scripts and complex scripts with program logic. Script commands are configured through the Properties screen.

There are two ways to open the Properties screen:

- Right click on a script command and select Properties from the drop-down menu.
- Double click on a script command, and the Properties screen is displayed.

The script commands described in this section are used in both Automation and Monitoring scripts, except where indicated. The Toggle Timers and Video Monitor (VM) commands are used exclusively in monitoring scripts.

DeviceAnywhere Studio helps you determine the correct name (command syntax) for each key on a device. Key names are found by hovering the mouse over a key, and the key name appears in square brackets. Enter these names including square brackets into the Text To Send field in order to define the key presses for a particular script command.

7.2 CheckPoint

All script commands have a tab to specify the command as a checkpoint. When the Checkpoint checkbox is selected in the Properties screen, all the images that appear on the device screen before and after a key press during script execution are captured.

Figure 7-1 illustrates the Checkpoint tab in the Properties screen.

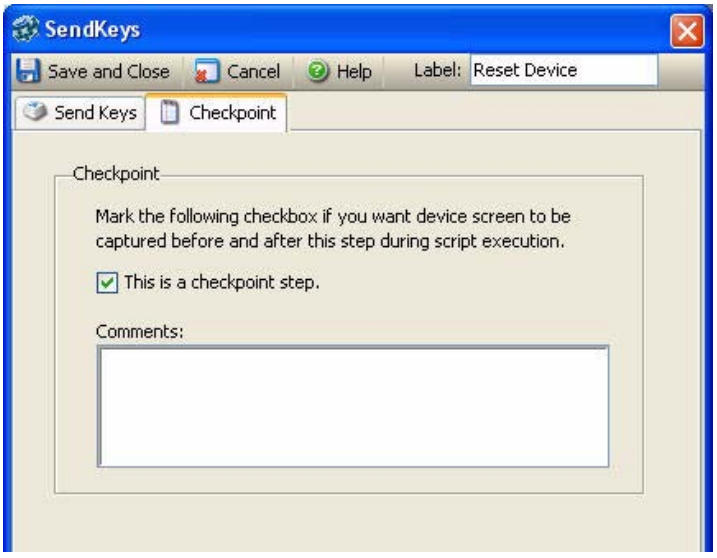


Figure 7-1 Checkpoint Tab

What are Checkpoints?

- Checkpoints are used to verify that your mobile application is functioning properly and serving the correct content. Checkpoints stop the execution of the test script temporarily. Without checkpoints in your script, you could send a series of key presses to the device in the SendKeys command, but you could never check to see whether you are getting the expected responses. Checking for expected responses is an important part of testing.
- Checkpoints are also important to help synchronize the test scripts. If you are testing network applications and you are uncertain of a transaction response time, you must include checkpoints that force the script to wait for the device to respond before moving to the next step.

The following script commands are available in *DeviceAnywhere*:

- Send Commands
  - SendKeys
  - Hardware
  - PlayAudio
  - FindAndTouch
- Wait Commands
  - Wait
  - WaitText
  - WaitImage
  - WaitEvent
  - WaitAudio
- Control Commands
  - NavigateTo
  - SetVariable
  - Branch
  - Loop
- Measure Commands (For Monitoring Scripts Only)
  - Toggle Timer
  - Video Monitor (VM)
- Extend Commands
  - Call Macro
  - Java Command
- Result Commands
  - Success
  - Fail

7.3 Script Command Descriptions

7.3.1 SendKeys Command

The SendKeys command allows you to execute key presses on the device using the Text To Send field. Text keys (alpha-numeric characters) are entered 'as is.' Special functions of the device, such as Power, Menu and Arrows, are entered using their key name in square brackets (e.g., [END]). For example, when you want to enter the number 555-1234 and click dial, you enter 5551234[SoftL] in the Text To Send field. Remember to remove any hyphens or spaces from the phone number in a Text to Send field.

The SendKeys command requires you to specify which of three key modes you are using. The three modes are: 1) Alpha, 2) Web, and 3) Numeric. The Alpha mode allows you to enter alpha-numeric characters. Some devices have a different key mapping when they are accessing the URL entry screen. The Web mode switches the key mapping to accommodate such a change. Use the Numeric mode when you are entering numbers into a field that specifies numeric entry only.

An additional attribute of the SendKeys command is it allows you to merge features of the Wait-Image command and specify the image to wait after each key press. Instead of using two commands, SendKeys and WaitImage, you can select the Use WaitImage checkbox in the Properties screen. An additional WaitImage tab then becomes visible. The SendKeys is executed first and then matched to the image extracted for the WaitImage.

7.3.2 SendKeys Properties Screen

Figure 7-2 illustrates the SendKeys Properties screen. The Properties screen in *DeviceAnywhere Studio* can change based on the product you are using. For example, the SendKeys Properties screen for Record & Playback is similar to the SendKeys Properties screen in Monitoring. However, their settings and appearance are somewhat different.

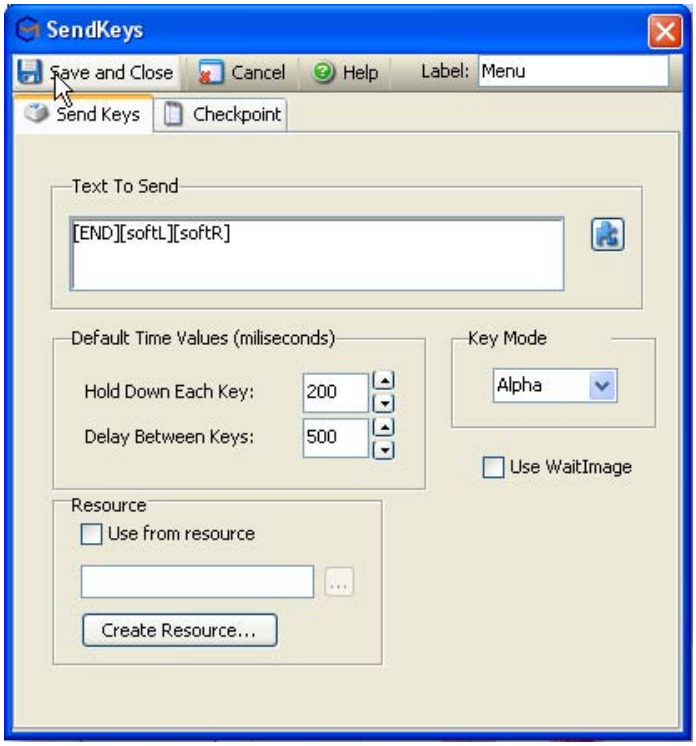


Figure 7-2 SendKeys Properties Screen

SendKeys configuration settings include entering data in the Text to Send field and setting up transmission characteristics. These settings are important in making a script execute to completion. For example, the Hold Down Each Key setting may need to be adjusted depending on network response time.

The SendKeys Properties screen has the same toolbar as the Script Properties screen. Refer to "Script Properties Screen" on page 93.

7.3.3 SendKeys Properties

The Properties tab (default view) provides the following settings:

- Text to Send—The Text to Send field contains the key press associated to the SendKeys. Insert Variable—The Insert Variable icon (puzzle piece) allows you to set Script variables.
- Hold Down Each Key—The Hold Down Each Key field lets you specify the amount of time in milliseconds the key press is to be held down.
- Delay Between Keys—This settings allows you to control the delay time between key presses.
- Key Mode—This field lets you select the key mode: Alpha, Numeric, or Web URL, from the drop-down menu.
- Use WaitImage Checkbox—When this box is checked, two new tabs, WaitImage and Timeout, are automatically available in the Properties screen.
- This is a checkpoint step Checkbox—When the checkpoint checkbox is selected, the images that appear on the device screen before and after the key press in the script execution are captured.

- Comments—Use this field to add comments about this SendKeys.

7.3.4 WaitImage Tab in SendKeys Properties

Figure 7-3 shows the WaitImage tab in the SendKeys Properties screen. This tab allows you to edit the reference image of the SendKeys command.

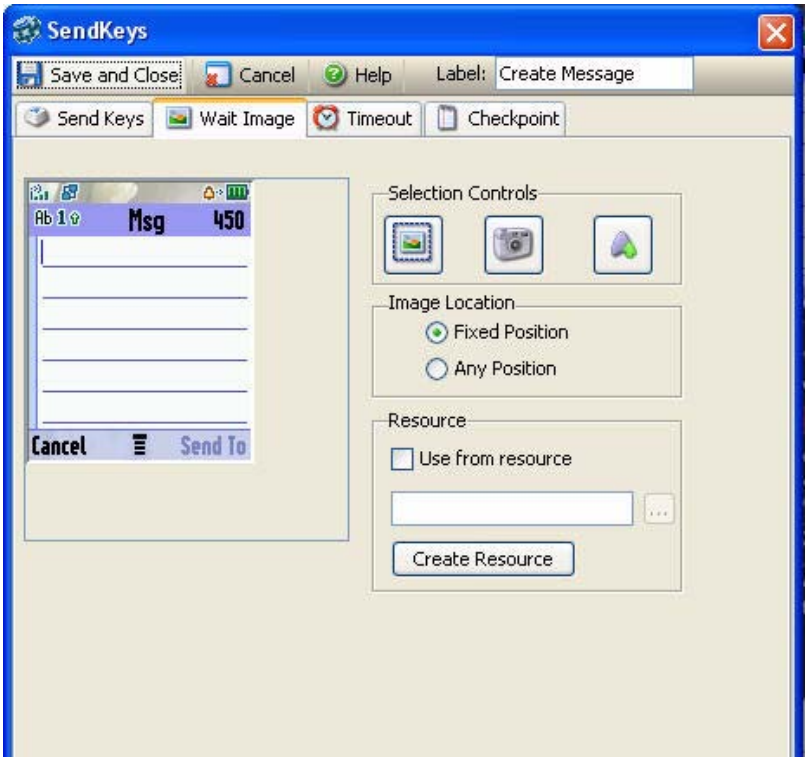


Figure 7-3 WaitImage Tab in SendKeys Properties

The WaitImage tab provides the following settings:

- Selection Control—This button is used to select a region of the reference image for the compare function. Click Selection Controls, then draw a rectangle on the image on the left to indicate the region that will be used in the compare.
- Camera—When the SendKeys is first added, the current device screen is captured. This reference image can be updated to the most current device screen by pressing the Camera button. If a new image is available, it will be displayed. You can pause the live recording, navigate to the correct frame by pressing the Device Video Controls, and return to live recording. This procedure allows you to enter any device screen that was previously recorded in the SendKeys command.
- Import—This option allows you to import an image from your desktop.
- Image Location—This indicator determines whether the compare function for the WaitImage command will search the image at a fixed position (exact coordinate location) or any position on the screen.
- Resource—These commands in the Resource window allow you to browse for an image in the Resource Manager or create a resource by storing the reference image from the SendKeys command. For more information on the Resource Manager, refer to "Resource Manager" on page 252.

You can import any image to *DeviceAnywhere* and use it for the comparison. In WaitImage, select the Import Image button. This action opens a browsing window where you can select any image in JPEG, JPG, PNG, GIF, DAT, or JPE format. After the selection is completed, the image selected is shown on the left side of the WaitImage tab. Now you can use this image for the comparison.

7.3.5 Timeout Tab in SendKeys Properties

Figure 7-4 illustrates the Timeout tab for the SendKeys command when Use WaitImage is checked. This tab allows you to specify the time before a timeout error occurs. To configure these settings correctly, you should anticipate network response time. The Timeout tab supports the Error Management system. It allows you to associate an Error Type to Error Categories and Subcategories that you define in the Error Management system. Error Management is set up as a prerequisite to configuring the Timeout tab.

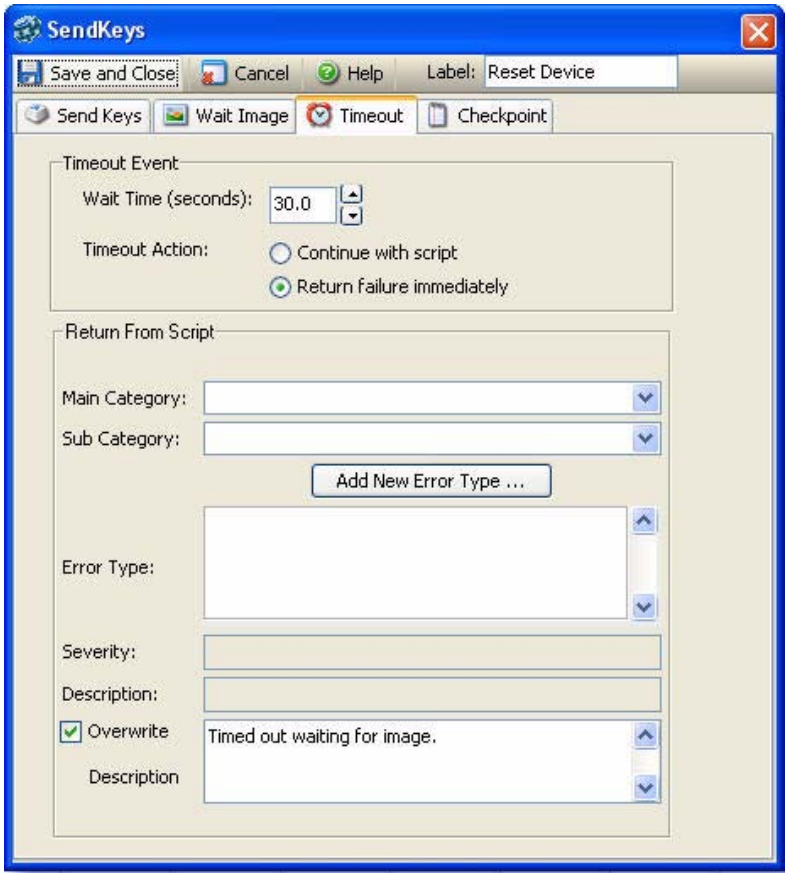


Figure 7-4 Timeout Tab

The Timeout tab has two windows:

- Timeout Event—This window allows you to set a timeout period in seconds before timeout occurs. It lets you set an action if a timeout occurs.
- Return from Script—This window allows you to specify an Error Type, Category, and/or Subcategory for the error condition. Severity refers to a rating from low to high used by the Error Management system. When the Overwrite box is checked, the Description field allows you to enter a custom error message that overrides the default error message in the field.

The Timeout Event window allows you to specify the following settings:

- Wait Time—Wait Time specifies how many seconds the command should wait for a defined action. For example while waiting for an image to download, you can add a timeout of 8 seconds. If the image does not load in 8 seconds, *DeviceAnywhere Studio* will declare a timeout.
- Time Out Action—Timeout Action allows you to specify the action to take in the event of a timeout. You can either 1) continue with the script, or 2) return a failure immediately.
  - In case of a timeout, the Continue with the Script option does not break script execution. The script continues with the next command. For example, while waiting for an image to load, if the command experiences a timeout, you may choose to continue with the rest of the script and execute the remaining commands.

– The Return Failure Immediately option allows you to stop script execution in case of a timeout. This may be required when the remaining script depends on the successful completion of a command. For example, if your script loads a WAP page, clicks on a link, and then performs tasks on the subsequent page, you can use the Return Failure Immediately option at the point where you click on the link. This terminates the script when the click on the link action times out.

The Return from Script window allows you to program error messages based on the custom Error Types, Categories, and Subcategories you define for Error Management. Error Management provides granularity for error reporting and handling.

Five script commands support Error Management:

- SendKeys with WaitImage checked
- WaitImage
- WaitText
- WaitEvent
- WaitAudio
- Fail

Use the drop-down menus on the Timeout tab to select the Error Type, Category, and Subcategory to configure the WaitImage command.

For a detailed description of Error Management, refer to "Error Management" on page 166.

7.3.6 Checkpoint Tab in SendKeys Properties

The Checkpoint tab controls the frame images captured before and after each checkpoint in the script. Check the box when you want the device screen images captured before and after the checkpoint. This tab also allows you to enter comments for the checkpoint.

Figure 7-5 Checkpoint Tab in SendKeys Properties

7.3.7 Hardware Command

The Hardware command is used to control the power and battery connection to a device. This command is useful in Reset Macros to initialize a device, since specific devices respond to this Hardware reset. Other devices respond to pressing the END key three times to reset correctly.

Figure 7-6 illustrates the Hardware command that is often used to reset devices.

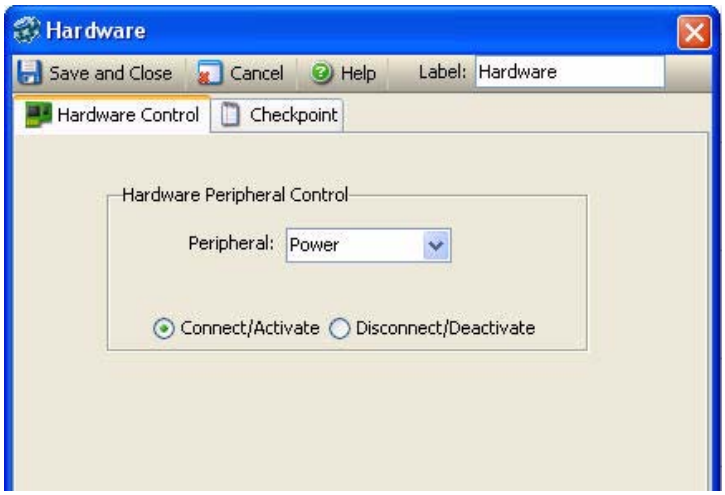


Figure 7-6 Hardware Control

7.3.8 PlayAudio

The Play Audio command is used to play a recorded audio into the microphone of the device. *DeviceAnywhere* supports three audio file types: AVI, MP3, and WAV.

Figure 7-7 PlayAudio

7.3.9 FindAndTouch

The FindAndTouch script command supports the execution of commands on a touchscreen device. The FindAndTouch command combines the functionality of a WaitImage and a SendKeys command. FindAndTouch compares a reference image to a graphic on a touchscreen device. It then captures the coordinates from a region of the screen that corresponds to the reference image. Similar to a SendKeys, The FindAndTouch command 'touches' the coordinates of the screen, thus selecting the appropriate menu item.

Figure 7-8 illustrates the FindAndTouch reference image that is selected for the compare operation.

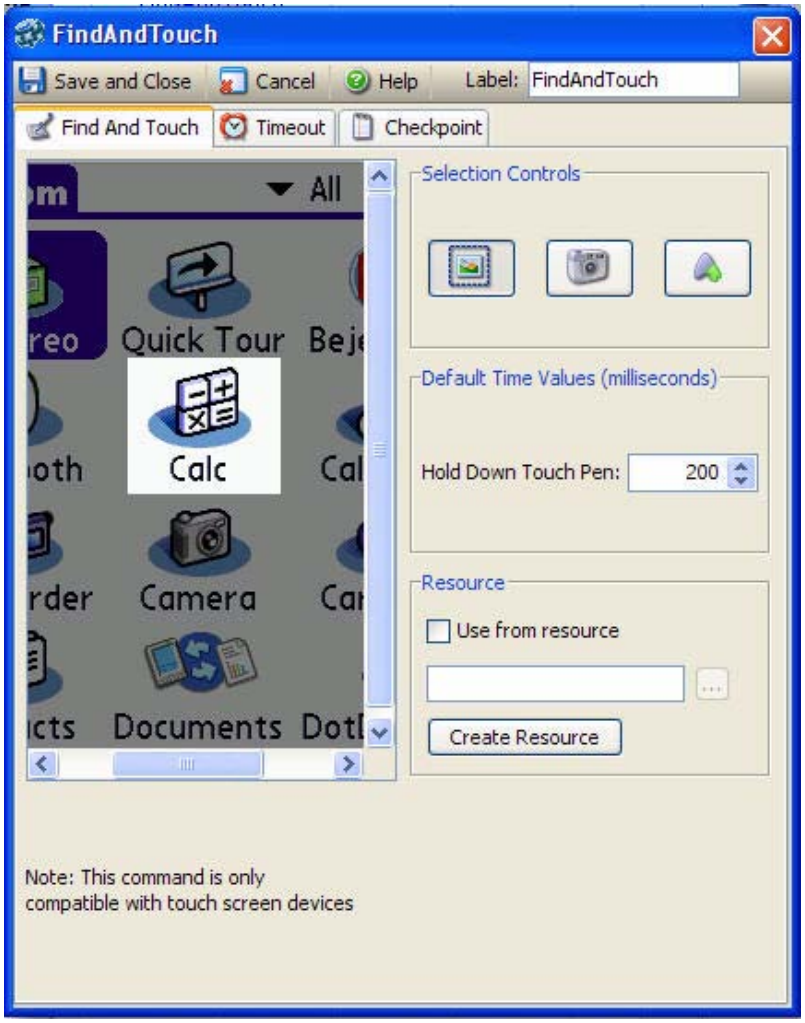


Figure 7-8 FindAndTouch Screen

FindAndTouch provides the following settings:

- Selection Control—This button is used to select a region of the reference image to capture coordinates of a touchscreen command. Click Selection Controls, then draw a rectangle on the image on the left to indicate the region that will be used in the compare.
- Camera—When the FindAndTouch is first added, the current device screen is captured. This reference image can be updated to the most current device screen by pressing the Camera button. If a new image is available, it will be displayed. You can pause the live recording, navigate to the correct frame by pressing the Device Video Controls, and return to live recording. This procedure allows you to enter any device screen that was previously recorded in the FindAndTouch command.
- Import—This option allows you to import an image from your desktop.
- Default Time Values—This option enables you to determine the hold down time for the touch pen.
- Resource—These commands in the Resource window allow you to browse for an image in the Resource Manager or create a resource by storing the reference image from the FindAndTouch command. For more information on the Resource Manager, refer to "Resource Manager" on page 252.

7.3.10 Wait Command

The Wait command allows you to specify how many seconds to wait before proceeding to the next command. This command is used to synchronize a script with device activity.

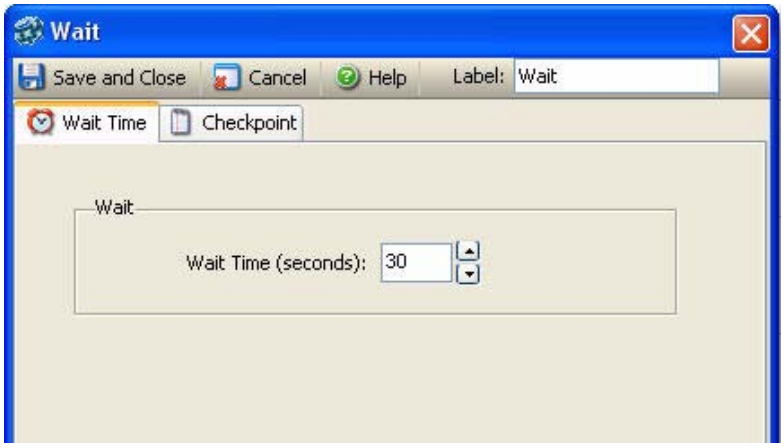


Figure 7-9 Wait Command

7.3.11 WaitText

WaitText is a conditional wait command. It waits either for a specified text string on the device screen or for a timeout when the text is not found. The same OCR capability is available in the WaitEvent command.

WaitText is based on the text matching capability of Abbyy FineReader OCR.

7.3.12 Using WaitText

Figure 7-10 shows the WaitText Properties screen.

Figure 7-10 WaitText Command

Briefly, WaitText provides the following windows:

- Text to Wait For—This window enables you to enter the text that qualifies the WaitText. When the text is found, the WaitText passes control to the next step.
- Extracted Text Sample—This window contains the extracted text from the current device screen. This screen enables you to review whether or not the extracted text is extracted correctly. For example, in Figure 7-10, the word Account is rendered as Accoi in the Extracted Text Sample.

- Resource—The commands in the Resource window enable you to browse for a Text resource in the Resource Manager or create a Text resource by storing the text from the WaitText command. For more information on the Resource Manager, refer to "Resource Manager" on page 252.
- Edit—The lower portion of the screen is used to facilitate text extraction by enabling you to modify the reference image of the device screen where text is extracted.
  - Capture Image enables you to take a new reference image for the text extraction.
  - Transform Image Before Text Extraction—When this box is checked, you have the option to modify the reference image in two ways: 1) Adjust contrast activates the slider bar at the bottom of the window. Thus, you can adjust the contrast of the image as needed; 2) Convert to Black and White activates the Eyedropper tool and transforms the image to black and white on the left.

When the WaitText command is first added, a reference image of the current device screen is stored. Then, the image is scanned for any text that appears on the screen.

When performing text extraction, make adjustment to the reference image. Adjust the contrast and sample the background color with the Eyedropper to get the most reliable text extraction as possible.

How to Perform Text Extraction by Sampling Background Color:

1. Drag a WaitText command into the script.
2. Ensure that you are using the correct reference image for the text extraction. If not, navigate to the device screen needed for the text extraction and click Capture Image.
3. Compare the text in the Extracted Text Sample window to the reference image.When the text matches, text extraction has completed successfully.
4. When the text does not match, select Transform Image Before Text Extraction.
5. Select Convert to Black and White.
6. Click the Eyedropper.
7. Touch the background color (contrast of text color) with the Eyedropper. In this case, touch the black background color. The word Accoi in the Extracted Text Sample window is updated to Account. This action indicates text extraction is successful.

Note: Text extraction is not 100% accurate. It can produce an error with similar characters (such as an I for a 1), since they are identical bit patterns in the fonts used for a device.

7.3.13 WaitImage Command

The WaitImage command waits for an image to appear or time out when the image is not found. The WaitImage command can wait for the entire screen to match, or can search for a region of the screen.

To clear the selected region, press the Selection Controls button again.

When a region of the image is selected, you can search for that region at any location in the image. Choose the Any Position radio button to scan the entire selected image for a match.

When selecting a reference image, choose a small portion of the screen that uniquely identifies the screen. Avoid the date and clock area. In this way, the comparison with the reference image completes successfully.

When the command is first added, an image of the current device screen is shown.

7.3.14 WaitImage Properties

Figure 7-11 illustrates the Properties screen for a WaitImage.

Figure 7-11 WaitImage Properties Screen

The WaitImage Properties screen has a number of settings to configure the WaitImage. It allows you to select an area of the reference image that will be used in the compare function. It also allows you to indicate where *DeviceAnywhere Studio* scans for the compare function.

How to Open the WaitImage Properties Screen:

To open the WaitImage Properties screen, double click on the WaitImage. The WaitImage Properties screen appears.

The WaitImage Properties screen has the same toolbar as the Script Properties screen. Refer to "Script Properties Screen" on page 93 for more information.

The following settings are available on the WaitImage Properties screen:

- Selection Controls—This button is used to select a region of the reference image for the compare function. Click Selection Controls, then draw a rectangle on the image on the left to indicate the region that will be used in the compare function. To discard the region, click the Selection Controls button again.
- Camera—When the SendKeys is first added, the current device screen is captured. This reference image can be updated to the most current device screen by pressing the Camera button. If a new image is available, it will be displayed. You can pause the live recording, navigate to the correct frame by pressing the Device Video Controls, and return to live recording. This procedure allows you to enter any device screen that was previously recorded in the SendKeys command.
- Import—This option allows you to import an image from your desktop.
- Resource—These commands in the Resource window allow you to browse for an image in the Resource Manager or create a resource by storing the reference image from the SendKeys command.

7.3.15 Set the Reference Image

By default, when a WaitImage command is inserted into a script, the entire screen is used for the compare function. The procedure below is useful in setting a region of the reference image in a WaitImage command. Comparing a small region of the reference image shortens processing time for the compare function.

How to Set the Reference Image:

1. Double click the WaitImage. The WaitImage Properties screen appears.
2. Click Selection Controls.
3. Draw a rectangle over the image (on the left) with the cursor to indicate the region of the image that will be used in the compare function. This included region is high-lighted.
4. Choose the location of the search: Fixed or Any Position (which scans the entire image for a match).
5. Click Save and Close.

The WaitImage is set to a region of the reference image for a compare function.

7.3.16 WaitEvent Command

The WaitEvent command is used to wait for an image, text extraction, or audio sample to appear on the device. A combination of these resources may also be selected. The WaitEvent command can branch the script into multiple paths depending on which image, text string, or audio sample was matched.

The Properties screen of the WaitEvent command allows you to add, remove and rearrange the image, text string, and audio sample being searched. Click on the Add New buttons to add a new search condition. A new tab appears, and a new path is added to the current execution path for the script.

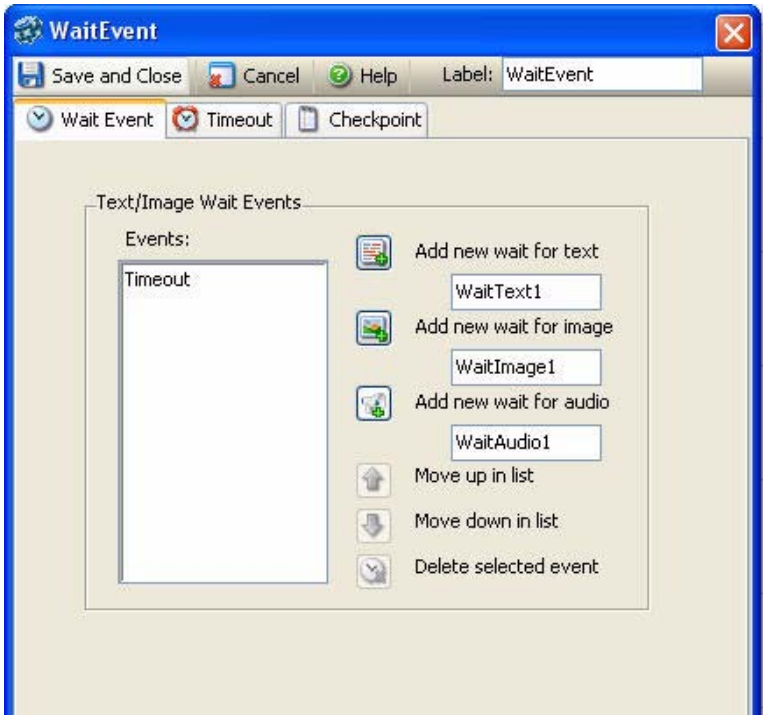
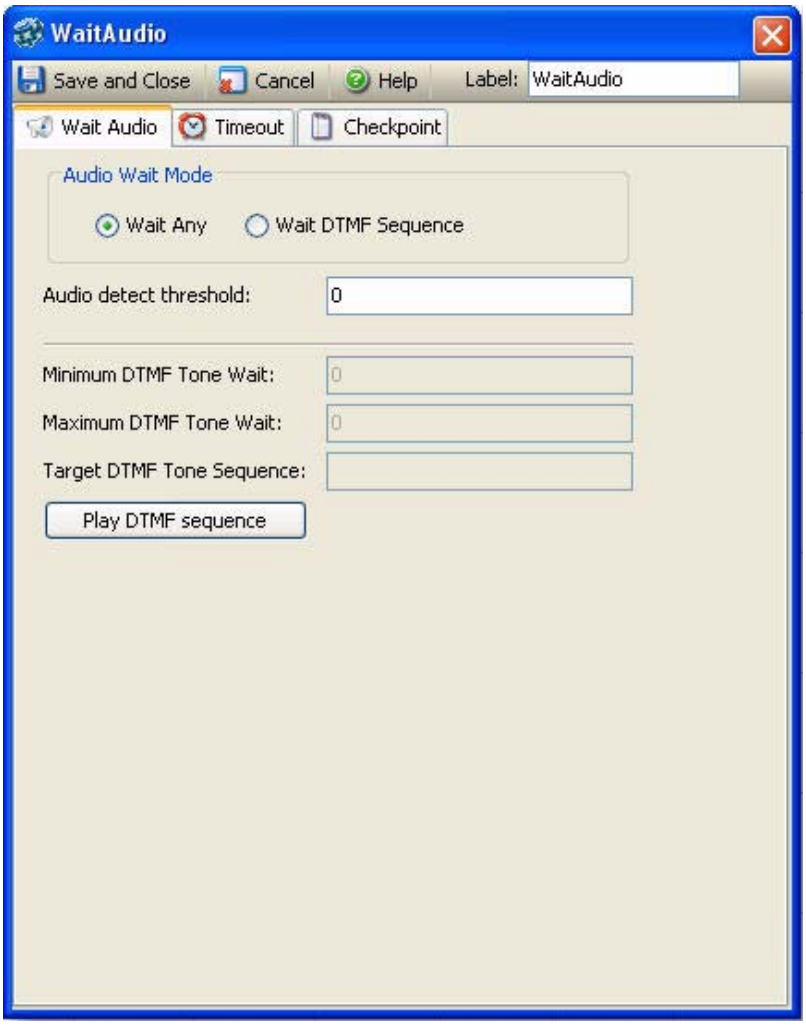


Figure 7-12 WaitEvent Command

Refer to the description of WaitText, WaitImage, and WaitAudio for further information on how to use these commands.

7.3.17 WaitAudio Command

The WaitAudio command is used to wait for an audio event on the device. For audio sample length, 20 seconds of audio is ideal for the compare operation. A maximum of 20MB is possible, but not recommended.



Wait Any is used to wait for any audio event to be detected by the device. It is not important what audio is being played, only that some valid audio was detected.

Set the Wait DTMF Sequence when you want to detect DTMF samples.

Enter the required Audio Detect Threshold. The Audio Detect Threshold represents audio volume. The Detection Threshold is a decimal value from 0 to 65534.

7.3.18 NavigateTo Command

The NavigateTo command allows a single key press to repeat until the specified condition in WaitImage/WaitText is met, or until the upper boundary of the loop is exceeded.

Set the NavigateTo branch by defining the key press in the Text to Send field and the upper boundary. The upper boundary corresponds to the number of iterations specified.

When configuring the WaitImage, you must select the target image that corresponds to the device menu screen that you are navigating to. The NavigateTo command automatically steps

through the device menu, stores key presses during each iteration until the target image is found and matched.

When the NavigateTo command is run, the SendKeys command is executed first, and then the text or image extraction is matched for each iteration. The command concludes the timeout when the image or text matches or the number of iterations exceeds the established upper boundary.

Figure 7-13 shows the NavTo Properties screen.

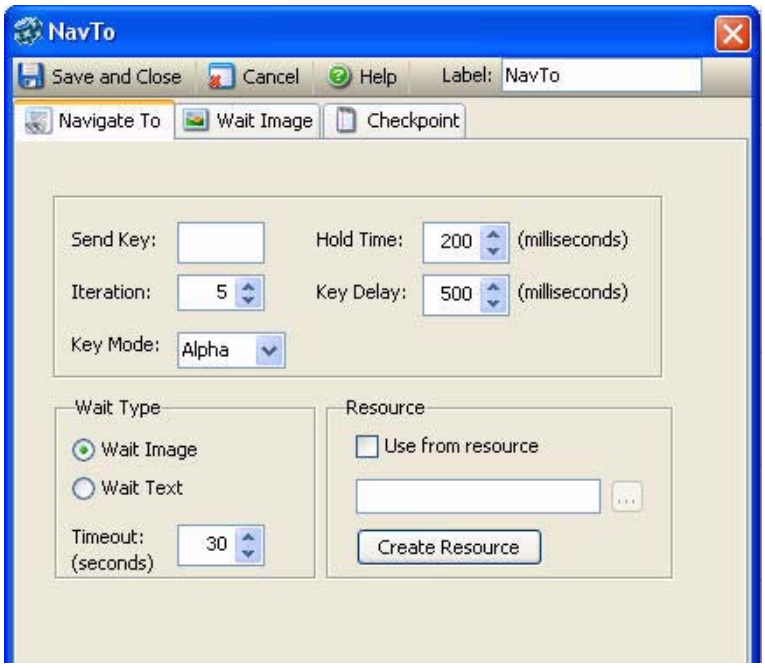


Figure 7-13 NavTo Command

Set the parameters in the NavTo Properties screen as described below:

- Send Key—Enter the key press to be repeated.
- Iteration—Choose the number of times to reiterate the key press from the drop-down menu. This number allows you to specify the exact number of times the key press is repeated, which will correspond to the key presses needed to navigate the device menus.
- Key Mode—Choose between Alpha, numeric, or web URL key modes.
- Hold Time—Enter the amount of time in milliseconds the key press is held.
- Key Delay—Enter the amount of time to elapse between key presses. This setting is useful to synchronize a script with device activity.

For Wait Type, select the specific condition that must be met to exit the NavTo command.

For Timeout, enter the amount of time for the compare operation to succeed before a timeout is declared.

The Resource window allows you to create Resources on-the-fly while building a script, or you can pull a Resource into the script from the Resource Manager.

Note that the NavigateTo command has two exit points: 1) success is when the text or image comparison is successful, and 2) failure occurs when the number of iterations exceeds the upper boundary, and image or text match is not found.

> Note: To continue on the Success path, insert a SendKeys command into the script specifying a [select] key press. This script command activates the highlighted image or text that was matched during the NavigateTo compare function.

7.3.19 SetVariable Command

The SetVariable command can be used to store a new value or update an existing value in a variable. The variable can be used in other parts of the script as part of branching and looping conditions or to replace text in strings being sent to a device. For example, a Boolean variable can be used to control a loop condition (e.g., loop until a value is set to true).

Figure 7-14 illustrates the SetVariable Properties screen.

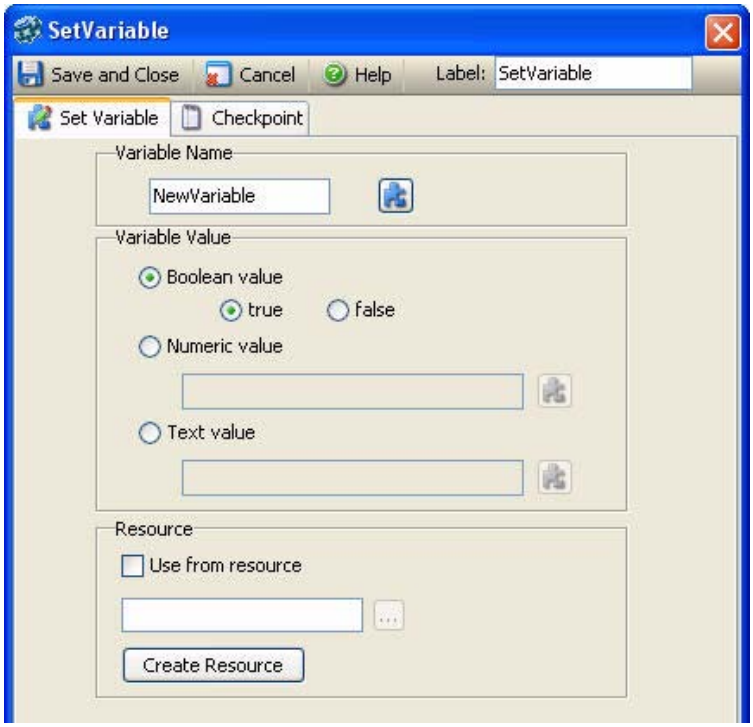


Figure 7-14 SetVariable Command

Variables enable users to change values without changing the script. For example, if there is a script that launches a browser with different Uniform Resource Locators (URLs), defining a variable called URL within the script allows you to re-use the same script to go to different URLs, by changing the value of the variable (the URL) without actually changing the script itself.

The Variable Name can be a new name (if this is the first time the value is being set) or can be chosen from one of the existing script variables or project variables. To select an existing vari-

able, click on the puzzle piece button. This button allows you to browse the variables in the Resource Manager. For further information, refer to “Resource Manager” on page 252.

If the variable name is new, it is added to the list of variables in the Insert Variable dialog and becomes available to other commands (such as branching and looping commands).

The variable value can be Boolean, Numeric or Text. All of these types are stored as text values, but different classifications are used to clarify the type of information expected in the variable.

Project variables (which are added under the Variables folder in the project) can also be updated with the SetVariable command. When these variables are updated, it does not affect other running scripts, but if this is done within a Macro script, the new value is visible to the calling script This procedure is one way to return values from a Macro.

Test Environments

Test environments extend the concept of variables so you can create different sets of variables, called environments. Each environment can have different variable values. For example, if your page has the URL http://dev.mc.com in your development environment, and the URL http://stage.mc.com is in your staging environment, you could define one variable called multi-URL and two test environments called Development and Staging. Within each environment, you would set the values for each variable. You can then run the same script, and depending on the environment you select, the respective value for the variables (e.g., multi-URL) are used.

7.3.20 Branch Command

The Branch command allows the control of the script to branch into multiple paths based on the value of one or more variables.

The Properties screen of the Branch command allows you to add, remove and rearrange the branches. Click on the Add new branch button to add a new branching condition. When you click on Add new branch, a new tab appears. And a new path is added to the current execution path for the script.

Figure 7-15 shows the Add New Branch command on the Branch Properties screen.

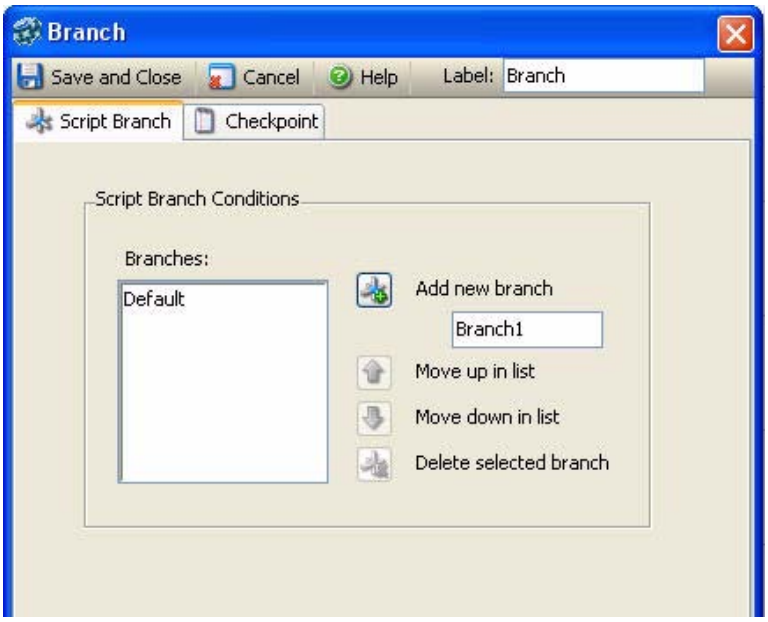


Figure 7-15 Branch Command

For each branch, a single variable expression or more complicated expression can be defined. In the variable expression, previously created variables (using SetVariable) can be inserted by clicking on the Insert Variable button.

Figure 7-16 illustrates the Branch1 tab where you can define a variable expression for the branch.

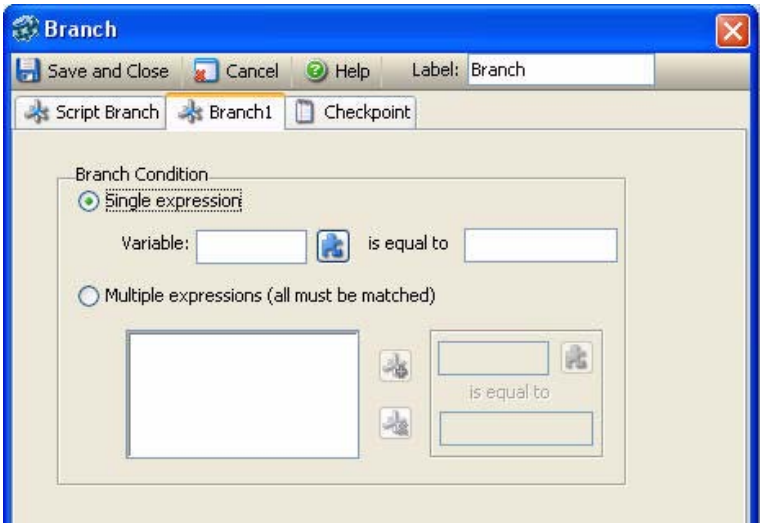


Figure 7-16 Variable Expression for Branch1

The variable is then compared with the value defined (for Boolean variables, it can be either true or false). When the match is successful, then the script control is transferred to that branch.

Branch Conditions

Branch conditions can be either single expression or multiple expression.

For a single expression, the single specified variable must match the specified value exactly (note for Boolean variables they will be either true or false). The matches are case sensitive.

For multiple expression branch conditions, ALL variables must match their specified values exactly. To add a new expression, select the variable, and click the Add button to include the variable in the expression list. The Remove button can be used to remove a selected expression.

Multiple Branch Matching

The new branches created in the script are the paths that are taken when either one of the expressions is matched or when none are matched, and the default path is taken. Each new branch appears as a red drop target where commands can be added. When the first command is added, the name of the branch appears next to the connecting link.

Note that if multiple paths match, the first one listed in the Properties screen is taken.

7.3.21 Loop Command

The Loop command allows a set of nested commands in the script to repeat a fixed number of times or until a condition is met with one or more variables.

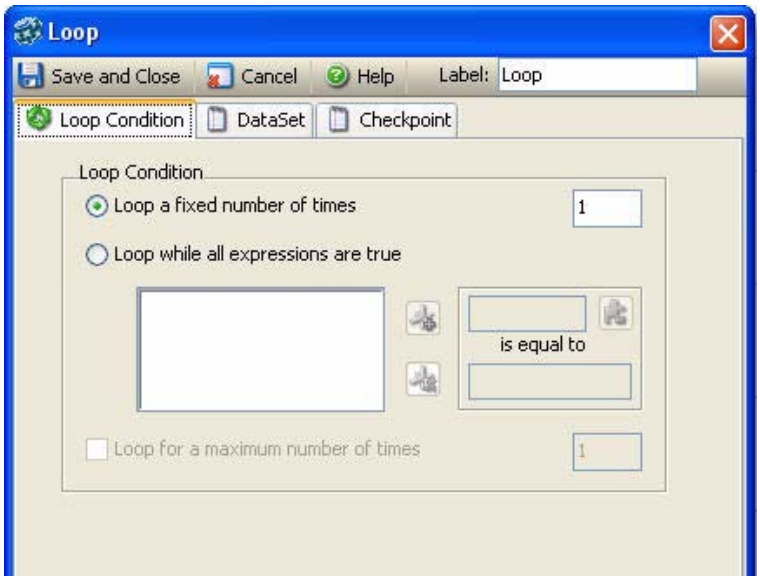


Figure 7-17 Loop Command

To loop a fixed number of times, simply specify the loop count in the Loop dialog box.

To keep the commands looping for multiple expressions, all of them must match the specified values exactly. As soon as any one expression fails to match, the loop exits. To add a new expression, select the variable and click on the Add button to include the variable in the expression list. The Remove button can be used to remove a selected expression.

The loop creates a new drop target where commands can be added and are nested within the loop (repeated each time the loop executes).

7.3.22 Toggle Timers

Toggle Timers are controls that measure time. Like a stopwatch, a Toggle Timer needs to be started and stopped. The time measured is used to calculate the success rate in the Production Monitor. To set up timing, drag a Toggle Timer into the script and configure it to start the clock. Then, after setting up intervening steps that you want to measure time on, drag another Toggle Timer into the script and configure it to stop the clock.

Each Toggle Timer is associated to a Timer in the Properties screen. The Timers tab is where you create a Timer. You can do this by entering a name and clicking the Add New Timer button. Each Timer is defined by a timer type.

7.3.23 Toggle Timer Properties Screen

The Toggle Timer is configured through the corresponding Properties screen.

How to Open the Toggle Timer Properties Screen:

Double click on the script command, and the Properties screen is displayed.

Figure 7-18 illustrates the Toggle Timer Properties screen. On this screen, you select Start or Stop for the Timer and select the Timer Type from the drop-down menu.

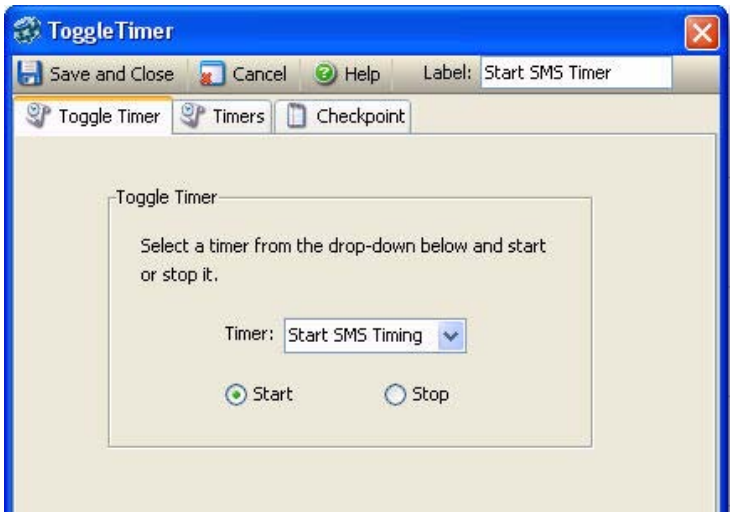


Figure 7-18 Toggle Timer Properties Screen

7.3.24 Creating Timers

Figure 7-19 shows the Timers tab that allows you to define, save, and delete timers associated to the Toggle Timer script command. The following Timer Types are available:

- One Time: A One Time timer reports only the time measured on the previous execution. For example, if a monitor script runs six times, a One Time timer measures the time interval six times, but only returns the last time recorded.
- Average: An Average timer returns the average time interval of all the recorded executions.
- Summation: A Summation timer returns the sum total of all the time intervals measured during the recorded executions.

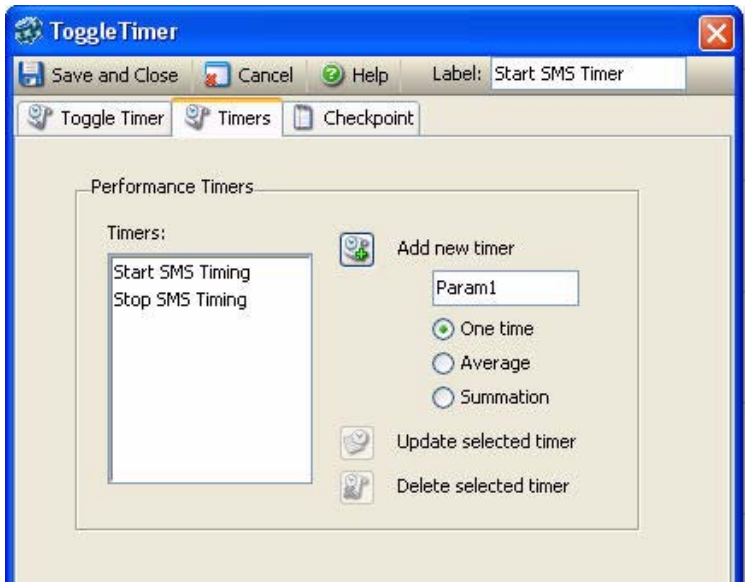


Figure 7-19 Timers Tab

7.3.25 VM Command

The VM command allows you to integrate video analysis tools into a monitoring script. VM commands measure video characteristics. Like a stopwatch, a VM command needs to be started and stopped. The Start VM and Stop VM commands are inserted before and after viewing a video clip. To set up video monitoring, drag a VM command into the script and configure it to start the video monitoring. Then, after setting up intervening steps to view a video clip, drag another VM command into the script and configure it to stop video monitoring.

Figure 7-20 shows the Properties screen for a VM Stop command. In this screen, the Save options are selected. The video will be saved in file ESPNNews at 12 frames per second and in AVI format.

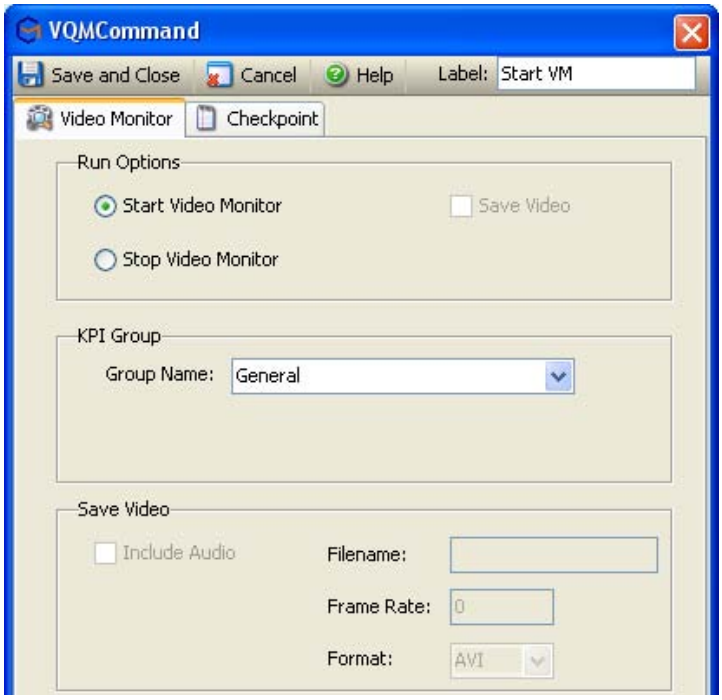


Figure 7-20 VM Command Properties Screen

Each VM command has a Properties screen that allows you to configure the following settings:

- Run Option—Start or Stop the Video Monitor
- KPI Group—Select the KPI group used as a basis to perform video analysis. Use the General setting as the default.
- Save Video—This option is used to save video in the Start Video Monitor VM command. The video can be save to a named file with a specified Frame rate and video format such as AVI, MP3, and WAV.

7.3.26 Macro Command

The Macro command executes the commands in a Macro before returning to the current script.

Unlike regular scripts, macros are device specific. When a Macro is called from a script, the Macro only executes when a Macro instance for the device matches the device in the calling script.

A Macro has a hierarchical structure which includes a Macro and a Macro instance. The Macro instance defines device-specific key presses.

Note that Macros can be associated with a different set of devices than their calling scripts. For example, a calling script may be associated with devices A and B, and a Macro may contain two Macro instances, one associated with A, and one associated with B. When the Macro is called, the correct Macro instance is chosen based on the currently selected device.

For example, if you have a script that performs two functions: 1) launches a browser, and 2) enters a URL, the process of launching a browser would differ for the Sanyo 8200 versus the Nokia 3650. In this case, a Macro is defined with two Macro instances: one for the Sanyo 8200,

and another for the Nokia 3650, each differing in the steps to launch the browser. When you call this Macro from a script using the Nokia 3650, the Macro instance for the Nokia 3650 is automatically selected.

Figure 7-21shows the Macro Initialize Device selected.

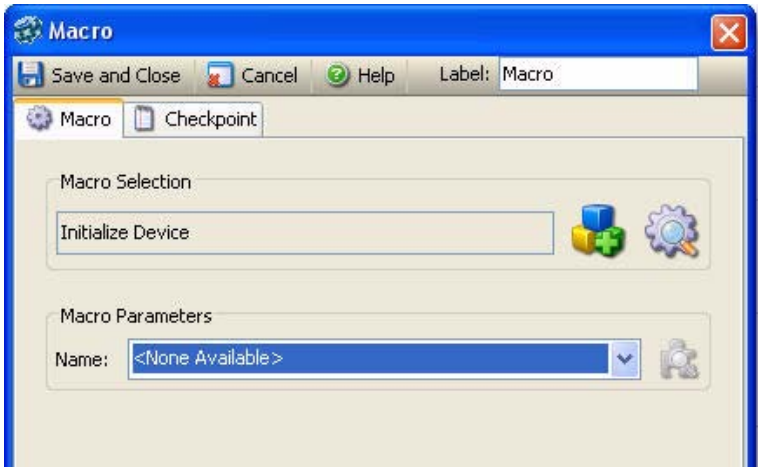


Figure 7-21 Macro Command

A Macro can be generalized with one or more parameters. For each parameter defined, the value to be passed to the Macro can be specified by selecting the parameter in the Script Parameters drop-down list. The default value as specified by the Macro will appear in the Value section of the page. This value can be overridden for this call to the Macro.

For further information on Creating Macros and Macro Libraries, refer to the Resource Manager.

7.3.27 Java Command

The Java Command allows Java code to be included in the test script. The Java Command has access to the current script variables and to the device interface that can be used to send key presses to a device or extract text from the screen.

Use the drop-down menu of the Insert field to insert Java Commands into the test script.

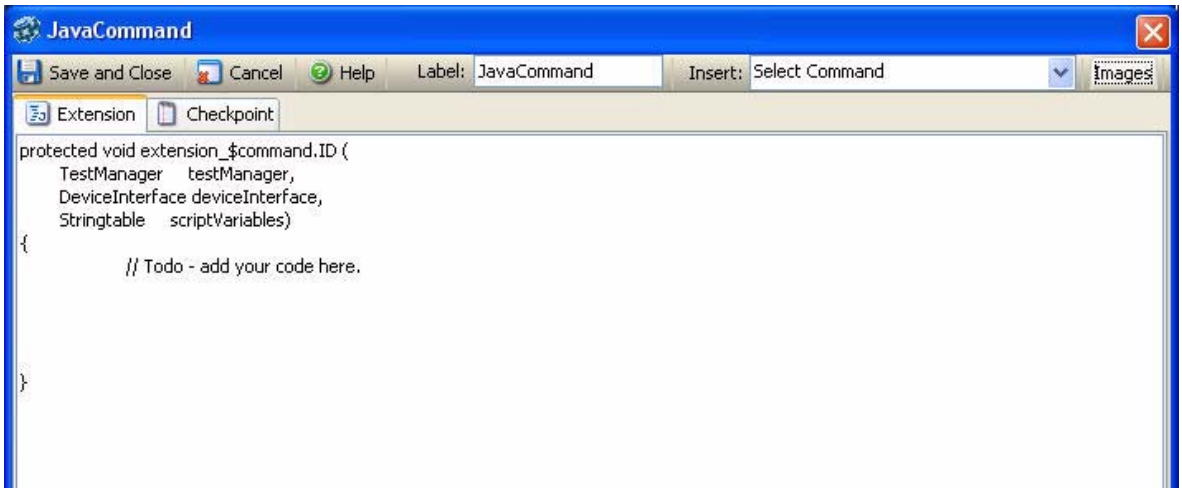


Figure 7-22 Java Extension

The Stringtable class supports the following commands for storing and retrieving variables in the script.

scriptVariables.put("Variable Name", "Variable Value");

String value = scriptVariables.get("Variable Name");

scriptVariables.putInt("Variable Name", intValue);

int value = scriptVariables.getInt("Variable Name");

scriptVariables.putBool("Variable Name", boolvalue);

boolean value = scriptVariables.getBool("Variable Name");

The DeviceInterface class supports the following commands for sending keys, and fetching text and image data from the device.

deviceInterface.sendKeys("Keys", "Key Mode", intHold-Time, intDelayTime);

String text = deviceInter-face.extractText(intForegroundColor);

int [][] image = deviceInterface.getSnapshot();

The following exception can also be thrown from the extension to halt the script execution:

throw new TestResult("Title", booleanSuccess, "Message");

7.3.28 Success Command

The Success command exits the script with a success condition. You can define the success message that is then stored as the final result of the script. When this command is used within a Macro Script, the Macro script returns to the calling script and execution continues.

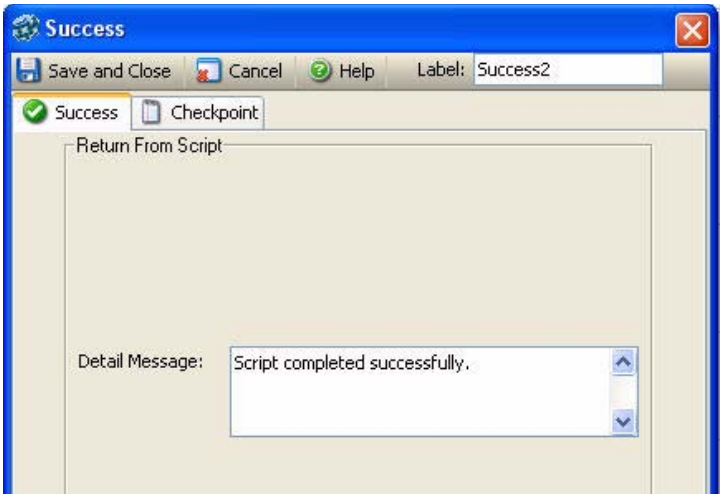


Figure 7-23 Success Command

7.3.29 Fail Command

The Fail command exits the script with a failed condition.

You can specify messages to be displayed associated with Error Types, Categories, and Subcategories that you defined in the Error Management system. The message is then stored as the final results of the script. This feature is supported in *DeviceAnywhere Monitoring*.

For further details on Error Management, refer to "Error Management" on page 166.

Note: When this command is used in a Macro, the entire script (including the calling script) exits immediately with a failure condition.

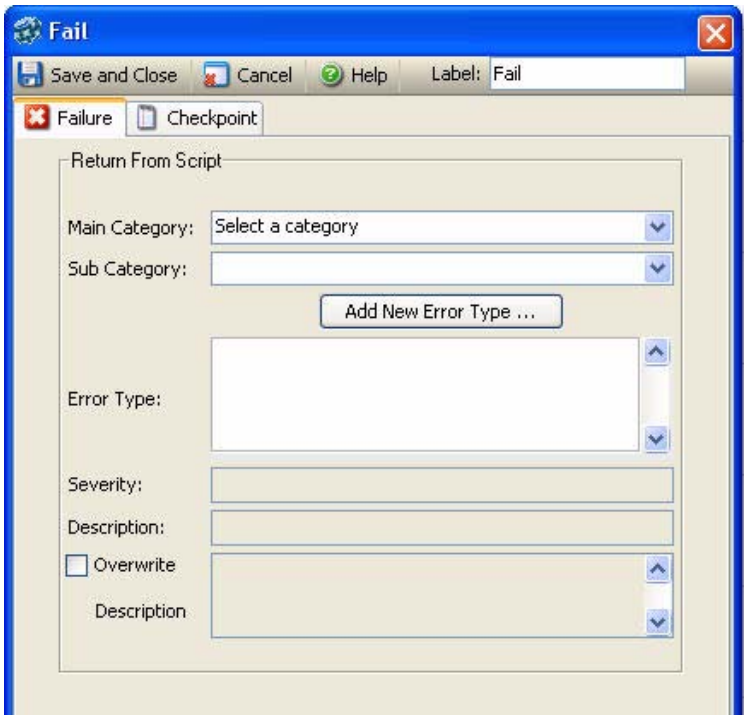


Figure 7-24 Fail Command

9 Resource Manager

9.1 Resource Manager Overview

The Resource Manager allows you to create libraries of resources. This feature is accessed by pressing the Resource Manager icon in the left Icon bar.

Figure 9-1 illustrates the Resource Manager view. The Alhambra library shows Image, Key, and Macro resources stored in the same library. The Macro library below it stores only Macros. The Resource Manager provides the flexibility to set up your resources in nested folders according to your project needs.

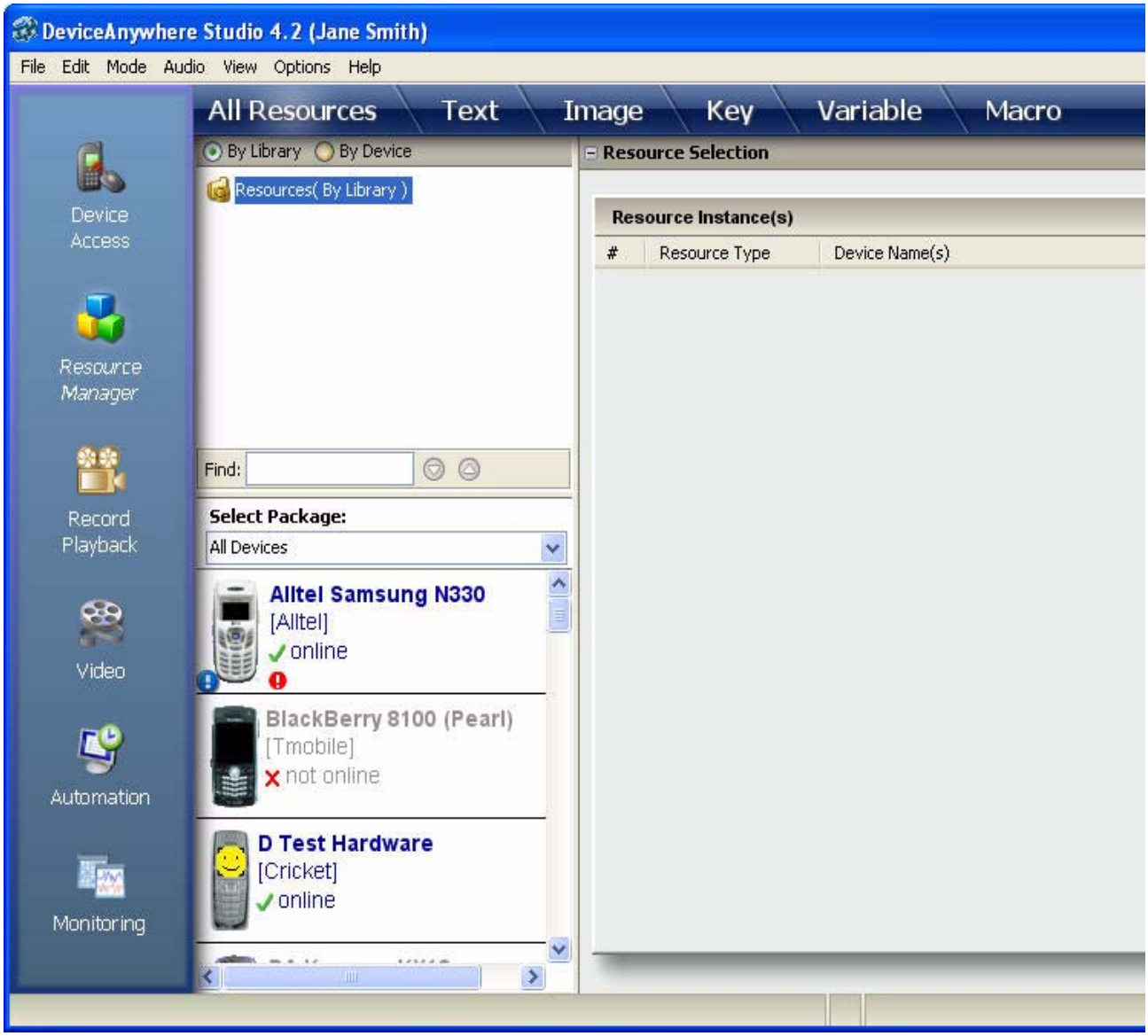


Figure 9-1 Resource Manager View with Libraries

The tabs in the toolbar include All Resources, Text, Image, Key, Variable, and Macro. Each tab corresponds to a resource type available in *DeviceAnywhere Studio*. Click on a tab to view and manage resources accordingly.

A brief description of these tabs follows:

- All Resources—This tab allows you to view and create any resource type. This is an excellent view to set up resource libraries as you start a project.
- Text—This tab allows you to manage and create Text resources only. When you are working in this view, you can still see all the library folders. However, you can only add Text resources.
- Image—This tab allows you to manage and create Image resources only. When you are working in this view, you can still see all the library folders. However, you can only add Image resources.
- Key—This tab allows you to manage and create Key resources only. When you are working in this view, you can still see all the library folders. However, you can only add Key resources.
- Variable—This tab allows you to manage and create Variable resources only. When you are working in this view, you can still see all the library folders. However, you can only add Variable resources.
- Macro—This tab allows you to manage and create Macro resources only. When you are working in this view, you can still see all the library folders. However, you can only add Macro resources.

Notice in Figure 9-1 the By Library and By Device indicators on the toolbar. The Resource Manager has two primary views.

When you select By Library, the libraries you've created are visible in the Explorer. You can expanded the folders in the Explorer to view and manage all the resources in the libraries.

When you select By Device, all the devices in the project are shown in the Explorer. You can expand the folders under each device to view and manage all the resources associated to each device.

9.2 What is a Resource?

In *DeviceAnywhere Studio*, a resource has a hierarchical structure. A resource includes a resource and multiple resource instances. For example, a resource is created to capture the idle screens for three *DeviceAnywhere* handsets.

Figure 9-2 shows Idle Screen, a Image resource. Three resource instances have been defined for Idle Screen. In this case, each resource instance stores the unique image that pertains to each device, respectively.

You are free to build your resources any way you like. However, for best results, you should try to create resources whenever you are planning to repeat the same action on multiple devices. By utilizing resources, you can quickly copy and paste your script to a new device and add new resource instances for that device, accordingly.

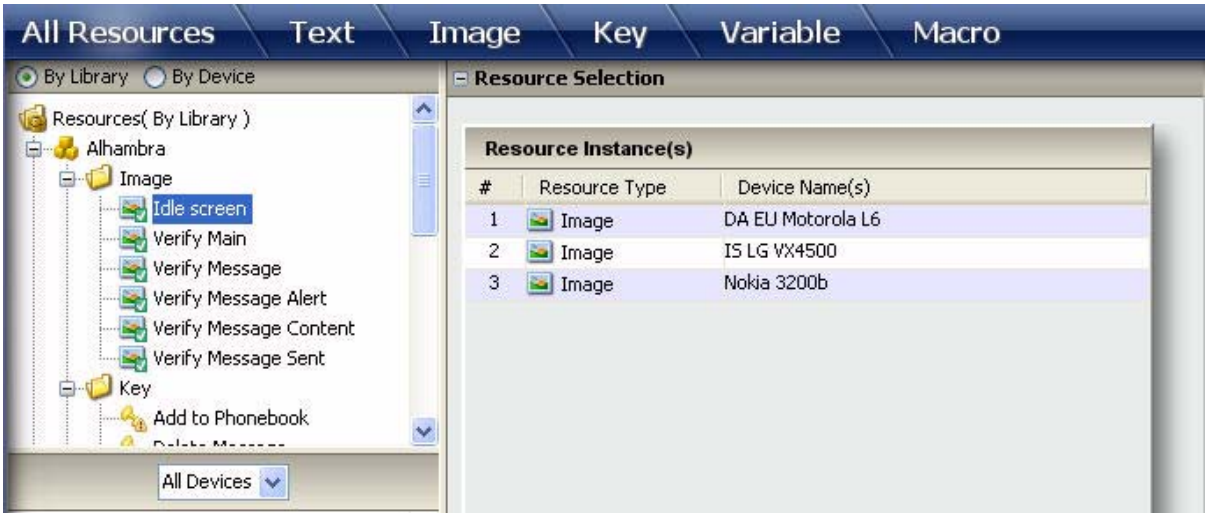


Figure 9-2 Idle Screen Resource Instances

9.2.1 Macro Resources

A Macro is a reusable mini-script that can be called from another script. Macros are commonly used actions such as 'initialize device' or 'go to URL' that can be in multiple test scripts for the same device. With Macros, you can write the mini-script once for a specific device and re-use it across multiple scripts.

Filenames for Macros must follow filenaming conventions:

- Use A-Z upper and lower case letters followed by numbers.
- Numbers cannot be used at the beginning of filenames.
- Java keywords are not permissible.
- Do not use the generic name TestScript.

Macros are like resources: they have a hierarchical structure. A Macro includes a Macro and multiple Macro instances. For example, if you create a Macro named Clear Cache, you could then create each of the Macro instances that will clear the cache for all of the devices in the test script. Each Macro instance may be entirely different. Creating individual Macro instances for each device allows the script to execute the proper key presses based on what device you associated with your test script.

Macros, unlike other resources, rely on script building techniques similar to those used in *DeviceAnywhere Pro* and *DeviceAnywhere Monitoring*. A section describing Macros in detail is presented later in this chapter.

9.2.2 Design Efficient Resources

Since resources allow you to test multiple devices in a script, you can design efficient resources by following some simple guidelines.

- Phones and smartphones are distinct in their key sequences. It is best to group devices with similar functionality into one resource. For example, group simple phones in one resource and more advanced smartphones in another.

- As a rule, devices activated on the same network exhibit similar functionality. Group similar devices on the same network into one resource.
- Group similar devices from the same manufacturer. For example, Motorola phones may share key sequence patterns that will help you develop a script more efficiently.
- Before you select a group of devices to use in a resource, run through the test steps on each device first. Compare key sequences between the devices, and then select the devices that share similar functionality.

9.2.3 Create Resources in Two Ways

In *DeviceAnywhere Studio* you can create resources in two ways:

- When you start a project, it is recommended to set up your libraries first. You'll need to set up libraries before being able to create resources on-the-fly in the script editor. When you set up the library initially, you can also create resources in the Resource Manager. It is ideal for developing a whole library very quickly.
- Once you've set up the libraries for a project, you can start designing your first script and create resources on-the-fly as you work in the script editor.

The following section describes how to set up a resource library including creating resources from within the Resource Manager.

9.3 Setting up a Resource Library

This section describes how to set up resource libraries including creating resources in the Resource Manager.

Setting up a resource library includes the following procedures:

- Add resource libraries
- Add category folders to the library to organize resources
- Create resources based on a resource type
- Create resource instances
- Implement the resource instances

9.3.1 Add a Resource Library

How to Create a Resource Library:

1. In the All Resources tab, right click on Resources (root) in the Explorer. A drop-down menu appears.
2. Select Add Resource Library. A new resource library appears in the Explorer.

A resource library can contain folders for all resource types. Figure 9-3 shows Add Resource library in the drop-down menu.

Figure 9-3 Add Resource Library

9.3.2 Add Category Folders

Category folders help you organize resources in a library. Category folders can be nested to many levels.

How to Create a Category Folder:

1. Right click on a Resource library. In this case, the Alhambra library.
2. Select Add Category. A new folder appears in the Explorer.

A folder can contain resources for all resource types. Figure 9-4 shows a new untitled folder added.

Figure 9-4 New Folder Added

9.3.3 Create a Resource

DeviceAnywhere Studio supports the creation of five resource types. Every resource is associated to a resource type when it is created. Each resource type is configured and associated to a script through a specific script command. For example, a Key resource is captured and associated to the SendKeys command, while an image resource is captured through a WaitImage command.

A brief description of each resource type follows.

- Text—A text resource is used to define a text string that you want to look for on the display of a device. It is used in a WaitText script command. Because each device may have a slight variation in displayed text or the OCR engine may not extract text consistently from device to device, it is important to be able to define the text and parameters used to identify specific text on a per device basis.
- Image—An image resource is used to define a reference image to look for on the display of the device. It is used in a WaitImage command. The image resource is particularly useful because each device has different screen characteristics. Therefore, WaitImage commands are not portable from device to device.
- Key—A key resource is used to define a sequence of key presses. It is used in a SendKeys command. A key resource is useful when navigating deterministic paths, such as device menus. Key sequences are usually quite short and are useful when timing or conditional logic is not required.
- Variable—A variable resource corresponds to alpha-numeric data, such as a name, telephone number, or text message. Variables allow scripts to be data driven. The contents of each field of a record in a data set is passed as a parameter to the script. Then, the script is executed repeatedly for each record in the data set.
- Macro—A Macro is a mini-script. It is composed of a small set of script commands and relies on script building techniques like test and monitoring scripts. The Macro allows you to control command execution by using the scripting commands.

How to Create a Resource:

1. Right click on a category folder. The drop-down menu appears.
2. Select Create a Resource Type.
3. Select Image, Text, Key, Variable, or Macro. Each resource type appears with an icon to identify the resource type.

A resource is generally given a generic name that represents a function or checkpoint used in a script, such as Reset (Macro), Go to Create Message (Key), or Verify Main Menu (WaitImage).

Enter notes about the resource in the Resource Description field.

9.3.4 Create a Resource Instance

A resource instance defines the unique key presses, references images, text, and so on, that belong to a specific device model.

How to Create a Resource Instance:

1. Select the resource in the Explorer.
2. Right click a device in the Device list.
3. Select Create a Resource Instance. A resource instance appears in the Resource Instance window.

The resource instance is created and associated to one device. You can add as many resource instances to a resource as you like.

When you want to create resources for more devices, simply right click on another device and choose Create a Resource Instance. Remember, you will need to acquire a device to record Wait-Images and to interact with the device for recording key presses.

Add Devices to a Resource Instance

The Resource Manager allows you to add additional devices to the same resource instance. This feature is used when you want to add multiple devices of the same model to a particular resource. For example, there may be more than one of the same model device in a particular package. Up to two additional devices can be added to a resource instance so you do not need to record the same resource twice.

How to Add Devices to a Resource Instance:

1. Select the resource in the Explorer.
2. Right click a device in the Device list.
3. Select Add to Resource Instance. A second device appears under the resource instance in the window.

Figure 9-5 illustrates the Add to resource instance command.

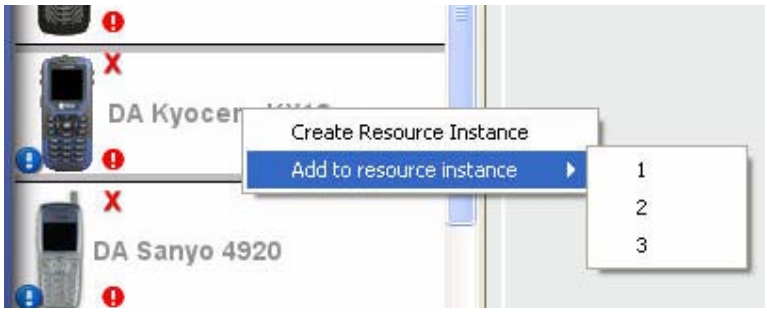


Figure 9-5 Add to Resource Instance

9.3.5 Implementing a Resource

Once you have created resource instances, you need to implement them. Implementing a resource means capturing the device screen, specifying a text string, or otherwise defining the resource instance. In the case of a Macro, you use the script editor to create a Macro script that runs to successful completion.

Figure 9-6 shows three resource instances that have not been implemented. Notice the Resource Type column. These resource instances have the Image resource type.

Resource Selection
Resource Instance(s)
Resource Type Device Name(s) Status
1 Image DA EU Motorola L6 Not Implemented
2 Image BlackBerry 8100 (Pearl) Not Implemented
3 Image JL Sanyo 7400 Not Implemented

Figure 9-6 Resource Instances (Not Implemented)

To implement a resource instance, click the resource instance in the Device Name column. The resource instance is context sensitive. An input screen appears to capture the device screen, specify a text string, or define the resource instance.

Image Resource Instance

In this example, a resource instance is created to capture the idle screen of a device.

How to Implement an Image Resource Instance:

1. Click the resource instance in the Device Name column. The WaitImage screen appears.
2. Using the device, navigate to select the reference image you want to capture. In this case, to capture a reference image for the idle screen, press the END key three times to return the device to the idle state.
3. Press the Camera icon. This action captures the image on the device screen.
4. Click Save and Close. The device screen is saved as the reference image.

9.3.6 Edit a Resource Instance

When you right click a resource instance, the Edit resource command is displayed. This command allows you modify the contents of the resource instance. Select Delete resource from the menu when you want to delete resources you don't need.

9.4 Resource Description

Every resource has a Resource Properties screen to help you keep notes on the resource. Right click the resource in the Explorer and select Resource Properties to open the Resource Properties screen.

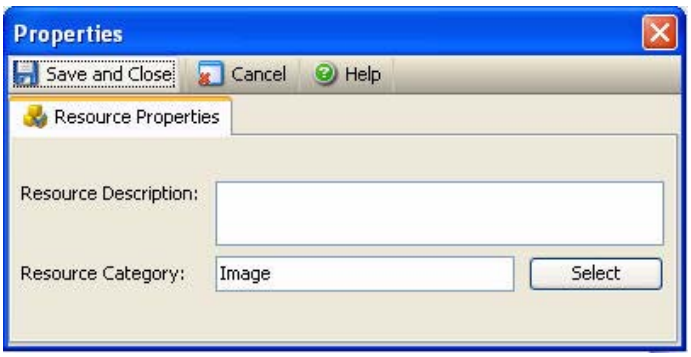


Figure 9-7 Resource Properties

9.5 Creating a Macro Library and Macros

A Macro is viewed and organized like other resources in the Resource Manager. First, create a Macro, then associate Macro instances for each device you would like to test.

To implement a Macro instance, use script building techniques like those described in “Automation” on page 77 and “Monitoring” on page 150.

The following is a set of guidelines for rapid Macro development:

- Use a combination of script recording techniques and object-oriented script-building to develop your Macro script.
- Control the key presses recorded into a SendKeys command by using Start Recording.
- Use copy (Control+C) and Paste from Clipboard features effectively.
- Use Wait commands when the script needs to synchronize with the device.
- Include WaitImages as 'checkpoints' in the script to synchronize with the device.
- Use the Run from Here command to execute the script from any point in the script in order to help save script development time.

For detailed information on script development, refer to Chapter 5 and Chapter 6. The procedures in the next section describe how to create a Macro library from start to finish.

9.5.1 Add a Resource Library

How to Add a Resource Library:

1. In the All Resources tab, right click on Resources (root) in the Explorer. The drop-down menu is displayed.
2. Select Add Resource Library. A new library is created in the Explorer.
3. Enter the name of the new library.

9.5.2 Add a Category Folder

How to Add a Category Folder:

1. Right click the new Resource library. The drop-down menu is displayed.
2. Select Add Category. A new category folder is created in the Explorer.
3. Enter the name of the folder.

9.5.3 Add a Macro

How to Add a Macro:

1. Right click on the category folder. The drop-down menu is displayed.
2. Select Create a Resource Type.
3. Select Macro. The Macro appears in the Explorer.

9.5.4 Add a Macro Instance

How to Add a Macro Instance:

1. Highlight and select the Macro in the Explorer.
2. Select a test device in the Device list.
3. Right click the device. A drop-down menu is displayed.
4. Select Create Resource Instance.

A new Macro instance is created in the Resource Instance window. Notice the icon in front of the Macro instance.

Figure 9-8 shows the empty script editor where you begin to build a new Macro.

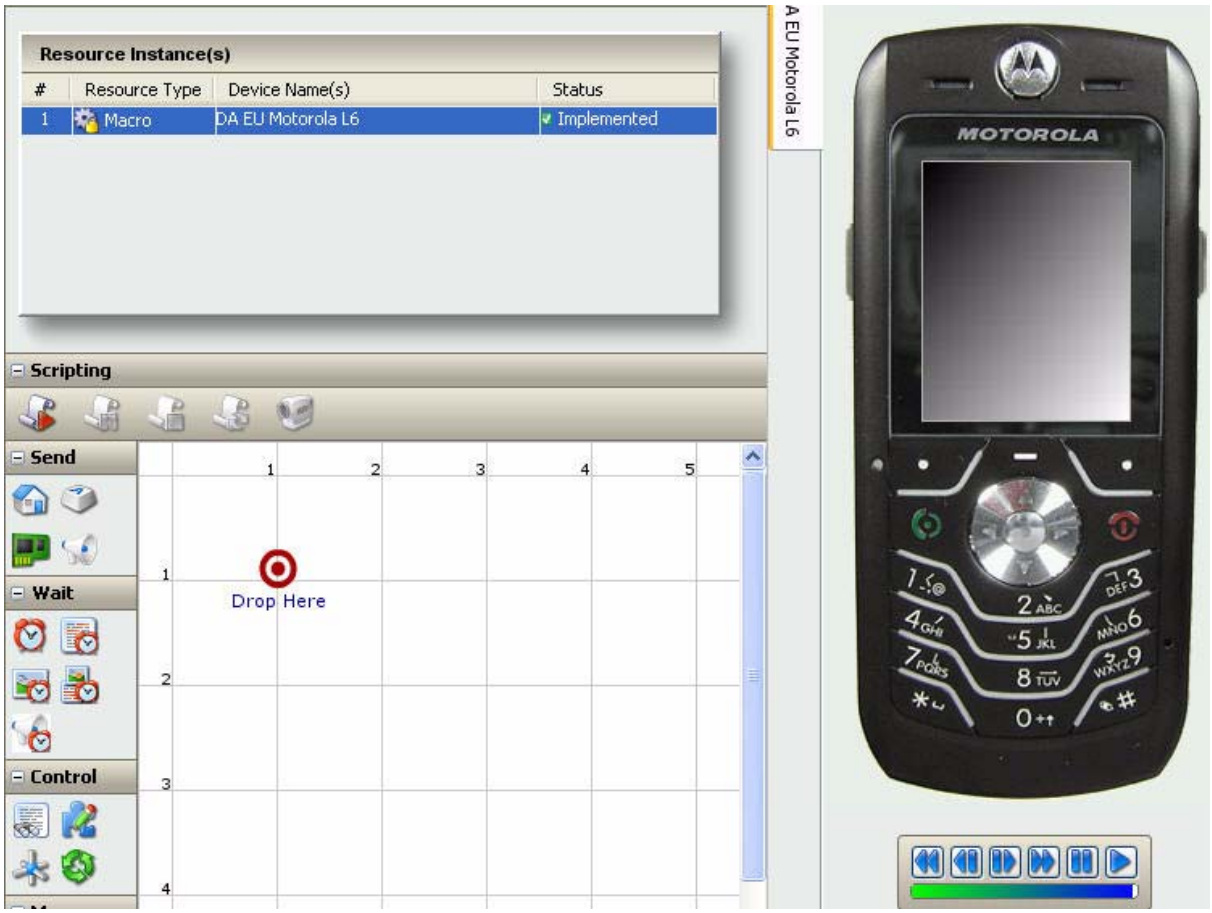


Figure 9-8 Macro Script Editor in Resource Manager

9.5.5 Access and Acquire the Device

How to Access and Acquire a Device:

1. Right click the device in the far right column. The Device menu is displayed.
2. Select Access Device.
3. Select Acquire Device.

9.5.6 Implement a Macro Instance

Before building a Macro, you should run through your test case on the device several times to determine what actions you want to incorporate into the Macro. Take note of any actions you may be repeating more than once. These repetitive commands can be turned into Macros so you do not need to script them multiple times. The following procedure provides an overview of creating a Macro. Use the examples that follow to start creating Macros for your application.

How to Implement a Macro Instance:

1. In the script editor, record key presses or drag and drop script commands into the script.
2. Test the Macro by executing the script or choosing the Run from Here command.
3. When the script runs successfully, right click the Macro instance in the Resource Instance window. A drop-down menu is displayed.
4. Click Save Changes.

Scripts and Macros are automatically saved to the database as you are working. However, Saving Changes writes an official copy of the Macro to the database for the script History. When you write the Macro to the database, the Lock icon is removed from the script or Macro to indicate that it is released for editing. Any team member can access the script or Macro at this point.

For both scripts and Macros, right click the script in the Explorer and select the Lock/Edit command to edit the script further after saving changes.

9.5.7 Manage a Macro Instance

Right click a Macro instance in the Resource Instance window. The drop-down menu provides commands to edit, delete, export, save changes, and so on, to the Macro instance.

Figure 9-9 illustrates the Macro Instance drop-down menu that is used to manage resources.

Figure 9-9 Macro Instance Drop-Down Menu

9.5.8 Macro Instance Examples

Initialize Device Macro

An initialize device Macro instance can be used at the beginning of just about every test and monitoring script. In script building, it is important to clear the device and set it in an idle state. In this way, all key sequences in the test script execute on the device as expected and allow observable results to be obtained.

An initialize device Macro can be simple, such as pressing the [END] or [Power] key three times or closing and opening a flip or slider phone. In most cases, the device returns to the idle state even when the device is deeply nested in menus.

An initialize device Macro can also be very elaborate, such as deleting all pending messages, clearing the cache, and otherwise clearing and adjusting settings to a known state. Smartphones generally require more initialization to navigate in a graphical interface before executing a test script.

Figure 9-10 shows the Initialize Device Macro instance for the LG VX4500. This device responds to a SendKey command that executes three END key presses and a hardware reset to exit Sleep mode. Therefore, Initialize Device for the LG VX4500 was designed with a SendKeys and Hardware Control command.

- The first SendKeys command executes three END key presses.
- The Hardware Control command sets Connect/Activate to wake up the device.

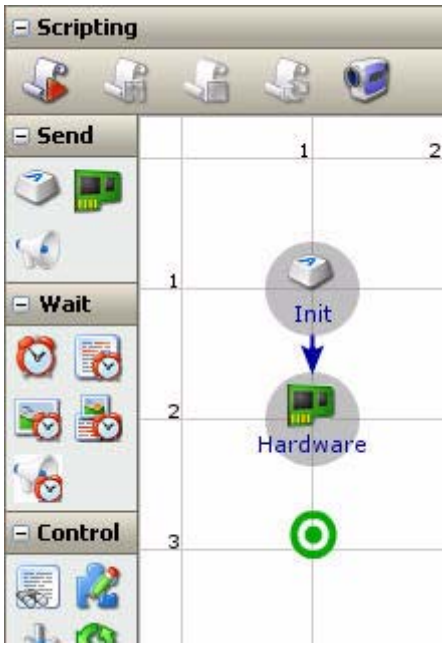


Figure 9-10 Initialize Device for LG VX4500

Table 9-1 Clear Cache Macro

Action	Step	Expected Results	Motorola L6
1. Init	Press the End key three times.	Device is reset.	[Power][Power] [Power]
2. Hardware	Drag Hardware Control command and configure to Power Connect/Activate setting.	Device returns to the idle screen.	Hardware Control Power Connect/Activate setting is selected.

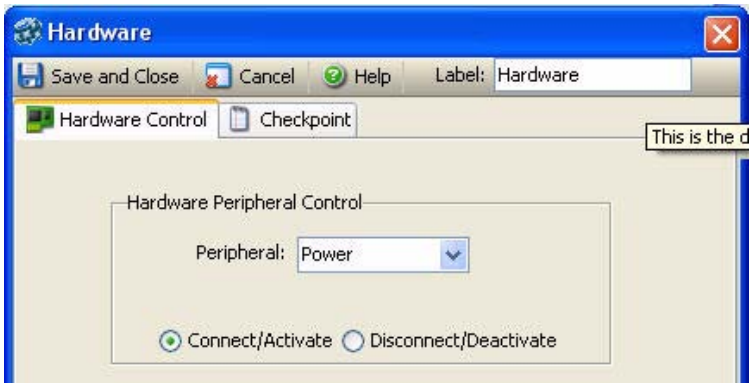


Figure 9-11 Hardware Control Settings to Initialize Device

Clear Cache Macro

Clearing the cache is an important function when you are measuring network download performance. Clearing the cache ensures that you are downloading a fresh browser page when you access a website.

Figure 9-12 illustrates the Clear Cache Macro instance in the script editor for the Motorola L6.

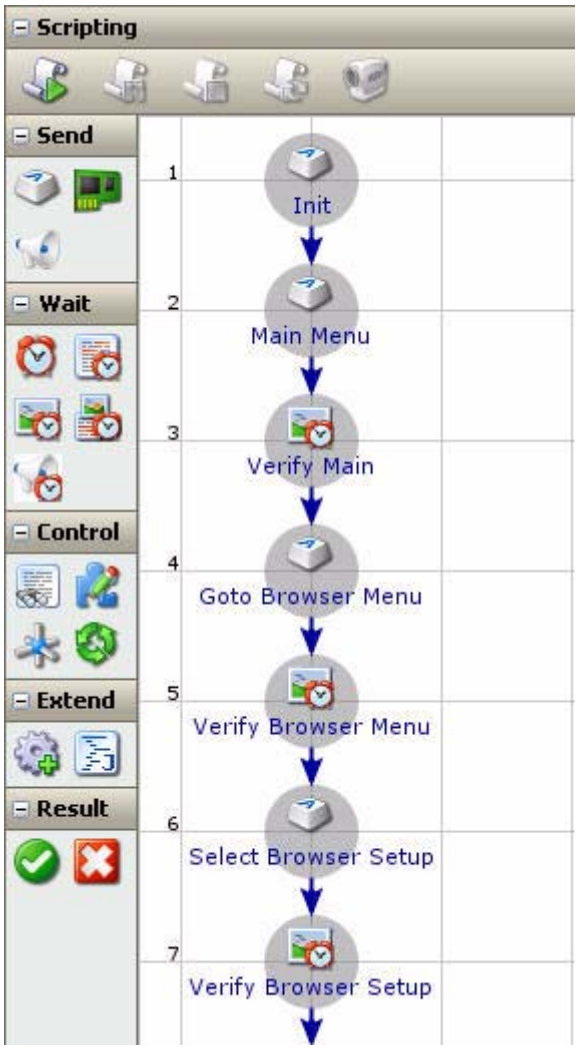


Figure 9-12 Clear Cache Macro

Table 9-2 provides a written description of the steps to clear the cache of the Motorola L6. The following key sequence is executed in the Clear Cache Macro for the Motorola L6. The commands in brackets correspond to command syntax entered in the Text to Send fields.

Table 9-2 Clear Cache Macro

Action	Step	Expected Results	Motorola L6
1. Init	Press the End key three times.	Device returns to idle screen.	[END][END][END]
2. Main Menu	Drag a SendKeys command and enter key presses to open the main menu.	The device main menu is displayed.	[Select]
3. Verify Main	Drag WaitImage and capture reference image for compare to the main menu.	Reference image for main menu screen saved.	WaitImage command succeeds.
4. GoTo Browser Menu	Drag a SendKeys command and enter key presses to navigate to the WebAccess icon.	The WebAccess menu is opened.	[Right][Select]
5. Verify Browser Menu	Drag WaitImage and capture reference image to compare to the WebAccess menu.	Reference image for the WebAccess menu screen saved.	WaitImage command succeeds.
6. Select Browser Setup	Drag a SendKeys command and enter key presses to select Browser Setup in the WebAccess menu.	The device navigates to the Browser Setup option.	[Down][Down] [Down][Down] [Select]
7. Verify Bowser Setup	Drag a WaitImage and capture a reference image to compare to the Browser Setup menu.	Reference image for the Browser Setup menu is saved.	WaitImage command succeeds.
8. Reset Cache	Drag a SendKeys command and enter key presses to navigate to Reset cache and execute Reset cache.	The Reset Cache option is selected and the cache is cleared.	[Down][Down] [Select]
9. Reset Device	Drag a SendKeys command and enter key presses to return the device to the idle state.	The device returns to the idle screen.	[SoftL][SoftL] [SoftL]

9.6 Bring a Resource into a Script

Let's assume you've created resources in the Resource Manager, but you haven't brought them into your test or monitor script. Make sure you are in the Automation or Monitoring view with the script editor open.

How to Insert a Resource into a Script:

1. Drag and drop a script command, such as a SendKeys command, into a script.
2. Double click on the script command to open the Properties screen.
3. Click the Use from Resource check box.
4. Click on the browser button to open the Select Key Resource window.
5. Click on the folders to navigate to a particular resource. In this case, select the Key resource.
6. Highlight the resource with the cursor.
7. Click Select. The name of the resource is entered into the Resource window in the Properties screen.
8. Click Save and Close.

Figure 9-13 shows the Reset Device resource selected from the Resource library and bound to the SendKeys command through the Properties screen.

Each script command is context sensitive. You can only select a Key resource to be bound to a SendKeys command, or an image resource to be bound to a WaitImage command, and so on.

Figure 9-13 Insert a Resource from the Properties Screen

9.6.1 Bring a Macro into a Script

For best practice, most scripts should begin with an Initialize Device Macro. It's easy to bring the a Macro into a script, once it is defined in the library.

1. Start by dragging a Call Macro command in the left panel to the top of the script.
2. Double click on the Macro command to open the Macro dialog box.
3. Click the Select icon on the right of the Macro selection window. The Select Macro Resource window is opened.
4. Click on the folders to navigate to a particular Macro resource. In this case, select Initialize Device.
5. Highlight the resource with the cursor.
6. Click Select. The name of the resource is entered into the Macro Selection window in the Properties screen.
7. Click Save and Close.

Figure 9-14 shows a Call Macro command dragged into a script.

Figure 9-14 Drag Call Macro into Script

Figure 9-15 illustrates the Macro dialog box where you select a Macro from a Resource library.

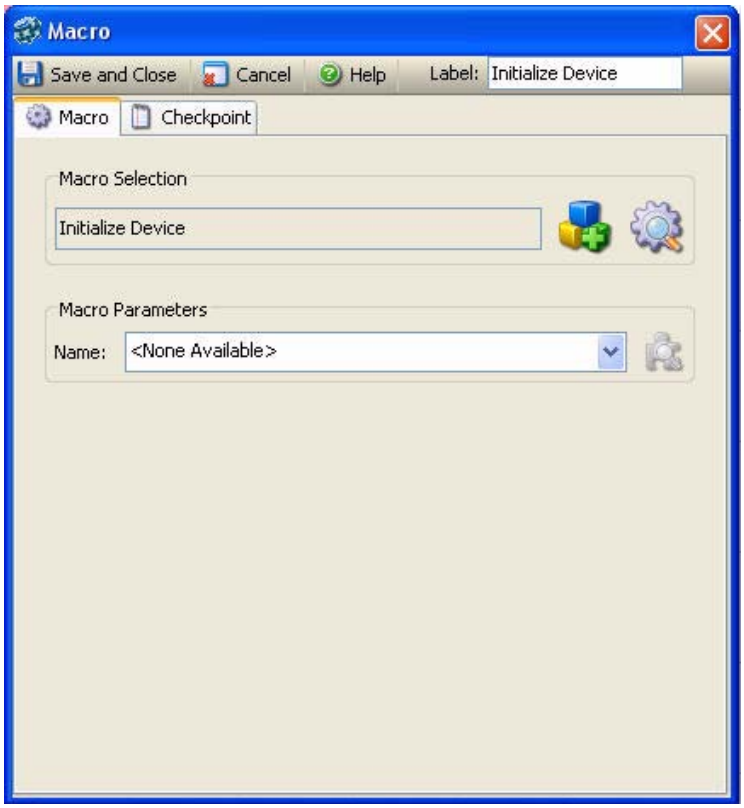


Figure 9-15 Macro Dialog Box

Save changes to the script when you want to check the version into the database. For Macros, right click the Macro instance in the Resource Instance window. Select Save changes from the drop-down menu.

9.6.2 Create a Resource On-the-Fly

All resources, except Macros, can be created on-the-fly and saved to a Resource library in the Resource Manager. In this example, you have already captured an image, a key sequence, or text in a script open in the script editor. Now it's a matter of saving the resource to a Resource library.

How to Create a Resource On-the-Fly:

1. Double-click on the script command to open the Properties screen.
2. Click on the Create Resource option. The Create Resource window is displayed.
3. Click on folders to navigate to the correct folder.
4. Enter the Resource name.
5. Click Save to save the resource in the Resource library.

Again the script command is context sensitive: you can only create resource types related to the script command you are working on. For example, you can create a Key resource from a Send-Keys command, and so on.

The benefit of creating resources on-the-fly is that it allows you to write your one test script, create resources on-the-fly for your first device, then go to the Resource Manager and create additional resource instances for the rest of the devices you want to test on. This is a quick way to get started building your resources so you can quickly copy a script and use it on other devices you have created resources for.

9.6.3 Create a Macro On-the-Fly

Macros rely on the copy and paste commands to copy a series of script commands from the script editor and pasting them into a valid Macro instance.

How to Create a Macro On-the-Fly:

1. In the script editor, use the cursor to draw a box around the script commands you want to include in the Macro. The commands are highlighted with a black circle.
2. Select Edit from the Menu Bar.
3. Select Copy Element. The script commands are copied to the Clip board. (Or use the keyboard shortcut Control+C.)
4. Go to the Resource Manager by clicking the icon in the Icon Bar.
5. Navigate to the folder where you want to store the Macro.
6. Right click the folder. The drop-down menu is displayed.
7. Select Create Resource Type.
8. Select Macro. A new Macro resource is created in the Explorer.
9. Enter the name of the new Macro.
10. Right click on the same device that is associated with the script you copied the script commands from. A new Macro instance is displayed in the Resource Instance window, and the script editor is opened with an empty script.
11. Select Edit from the Menu Bar.
12. Select Paste Element. The script commands from the script editor are pasted into a valid Macro instance.
13. Modify the Macro script as needed.
14. When you are done, save changes.

To save changes to a Macro, right click the Macro instance in the Resource Instance window and select Save Changes.

9.7 Designing Scripts with Resources

This section covers advanced topics in building scripts with resources. Resources provide a level of abstraction between the test or monitoring scripts and the devices that will be tested.

A script with resources is usually more compact in comparison to the end-to-end scripts you've seen in previous chapters. As you look over a script that uses resources, you'll see initialization, steering, Macro functions, checkpoint, and exit logic. However, most of the important key sequences that execute functions on a device are implemented at the resource instance and Macro level.

9.7.1 Send SMS Message with Resources—An Example

In Chapter 5 Automation, a Send SMS Message script is described in detail. The Send SMS Message script was created initially for only one device.

In this section, the Send SMS Message script is revised. It includes Image, Key, and Macro resources so that it will run with the following three devices:

- Motorola L6
- LG VX4500
- Nokia 3200b

The heart of this script is the Send Message Macro. It is defined with three resource instances to capture the unique key sequences to send an SMS message on the three devices, respectively. The Motorola L6 and Nokia 3200b devices use a Send Message key sequence that prompts for the SMS message followed by the phone number. The LG VX4500 prompts for the phone number first followed by the SMS message. The script executes each key sequence correctly, since the difference between key sequences for each device is handled at the resource and Macro level.

9.7.2 Script for Send SMS Message

A successful script breaks up key sequences into nodes that identify a progression in the tasks being performed. For example, the Send SMS Message could be composed of three SendKeys: one each for Initialize Device, Send Message, and Reset Device. This type of script does not offer much granularity between devices. However, the script shown in Figure 9-16 and Figure 9-17 breaks up key sequences in such a way that the underlying resources and Macros identify the unique key sequences between devices.

Figure 9-16 and Figure 9-17 illustrate the steps in the Send SMS Message script.

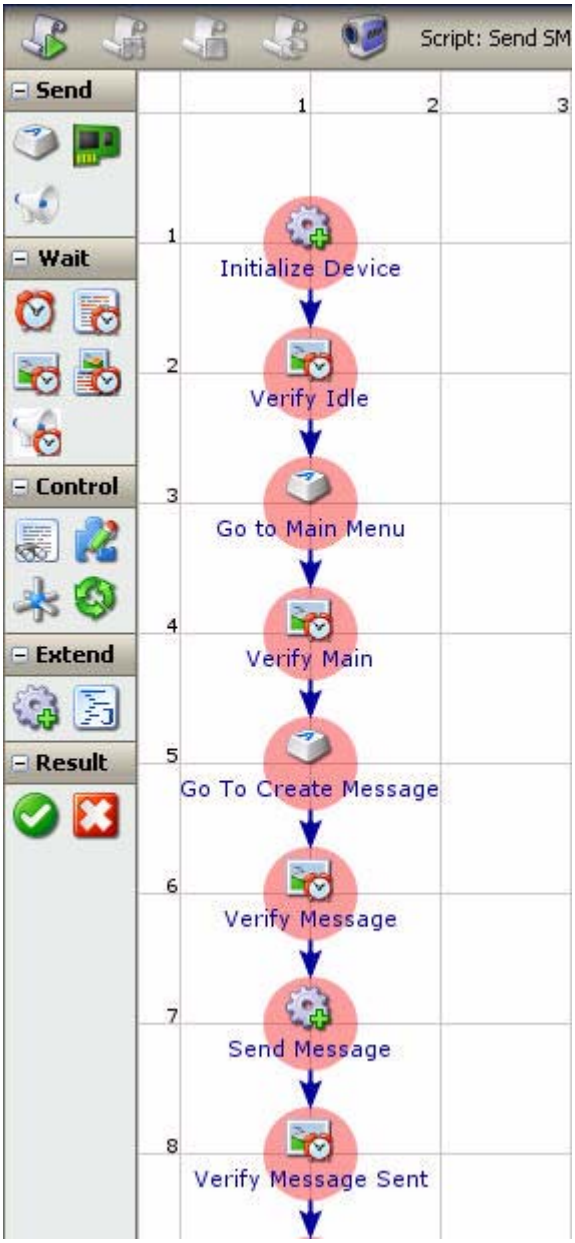


Figure 9-16 Send an SMS Message (Top)

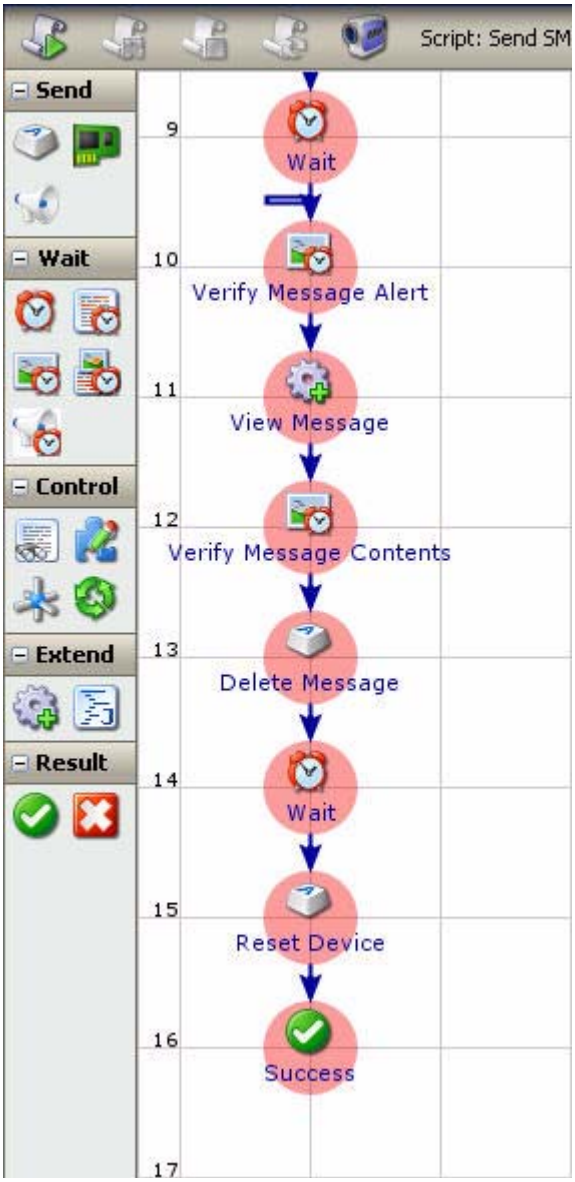


Figure 9-17 Send an SMS Message (Bottom)

The steps in the script with key sequences for each device are given in Table 9-3.

Table 9-3 Send an SMS Message

Action	Expected Result	Motorola L6	LG VX4500	Nokia 3200b
Initialize Device	Device is active and in the idle state.	[END][END][END]	[END][END][END] Hardware Control Connect/Activate	[END][END][END]
Verify Idle	Checkpoint to verify the device is in the idle state.	WaitImage command succeeds.	WaitImage command succeeds.	WaitImage command succeeds.
Wait	Synchronize script to device before going to main menu.	Pause in script	Only the LG VX4500 needed a pause in the script to synchronize with the device.	Pause in script
Go to Main Menu	Go to main menu.	[Select]	[SoftL]	[SoftL]
Verify Main	Checkpoint to verify the device is in the main menu.	WaitImage command succeeds.	WaitImage command succeeds.	WaitImage command succeeds.
Go to Create Message	Go to create message screen.	[Up][Right][Select][Select][Select]	[Down][Down][Select][Down][Select][Select]	[SoftL][SoftL][SoftL]
Verify Message	Checkpoint to verify the device is in an input state to create a message.	WaitImage command succeeds.	WaitImage command succeeds.	WaitImage command succeeds.
Send Message	Execute key presses to enter a SMS, type a telephone number, and send the message.	Three commands 1. Wait 2. “Automated SMS message.” 3. [SoftR][Select]6506555555[SoftR][SoftR]	Three SendKeys 1. Wait 2.6506555555 [Select][Select] 3. Automated SMS message. 4. [Select][SoftL]	Five SendKeys 1. Automated SMS message. 2. [SoftL][SoftL]6506555555[SoftL] 3. Return to Idle [SoftR] 4. Wait 5. [SoftR]
Verify Message Sent	Checkpoint to verify the device sent a SMS message.	WaitImage command succeeds.	WaitImage command succeeds.	WaitImage command succeeds.

Table 9-3 Send an SMS Message (Continued)

Action	Expected Result	Motorola L6	LG VX4500	Nokia 3200b
Wait	Synchronize script to device before receiving a Message Alert.	Pause in script	Only the LG VX4500 needed a long pause in the script to synchronize with the device.	Pause in script
Verify Message Alert	Checkpoint to verify the device received a new SMS message.	WaitImage command succeeds.	WaitImage command succeeds.	WaitImage command succeeds.
View Message	Open Inbox	[SoftR] The Hold down time for each key was changed to 300.	[Select]	Two SendKeys 1. [SoftI][SoftL] 2. [SoftL]
Verify Message Contents	Checkpoint to verify the device is in an open message.	WaitImage command succeeds.	WaitImage command succeeds.	WaitImage command succeeds.
Delete Message	Select option to delete the message.	[Menu][Menu][Select]	[SoftR][Down][Down][Select][Up][Select]	[SoftL][SoftL][SoftL][SoftL]
Wait	Synchronize script to device before going back to idle screen.	Pause in script	Only the LG VX4500 needed a short pause in the script to synchronize with the device.	Pause in script
Reset Device	Return to idle screen.	[SoftL][SoftL]	[Power]	[SoftR]

9.7.3 Key Sequence Comparison

In this section, you'll comparing the key sequences between devices to give you an idea of the trade-offs needed to create a multi-device script with resources. The four functions compared here include the following:

- Initialize Device
- Go to Create Message
- Send Message
- Reset Device

Initialize Device

At the start of every script, it is recommended to set the device to an active state and showing the idle screen. For many devices, such as the Motorola L6 and the Nokia 3200b, the execution of three END key presses is sufficient to bring the devices into an active state. However, the LG

VX4500 is a flip phone and can be set to an active state by closing and opening the flip phone by using the Hardware Control command.

Since Initialize Device requires more than a simple set of keys to initialize a device, the Initialize Device function is implemented as a Macro. The Initialize Device Macro is defined with three Macro instances, one for each device tested in the script.

- The Motorola L6 and Nokia 3200b Macro instances have one SendKeys that executes three END key presses.
- The LG VX4500 Macro instance has one SendKeys that executes three END key presses. In addition, a Hardware Control command was dragged into the script. The Hardware Control command was set to Connect/Activate the flip phone. In this way, the flip phone is properly initialized and responds to script execution.

Go to Create Message

For each of the three devices, the Go to Create Message navigates the device menu to reach the message input screen. The Go to Create Message is implemented as a Key resource with three resources instances.

Send Message

The Send Message function includes entering a text message, a telephone number, and then executing the Send command. The Motorola L6 and the Nokia 3200b share similar behavior: The SMS entry field appears first and is followed by the telephone number field. In contrast, the LG VX4500 displays the telephone number field first and is followed by the SMS entry field.

Since the Send Message function is implemented by a Macro, the differences between the key sequences have no impact on script execution. At the Macro level, it is also possible to synchronize the script based on device behavior.

For example, the Motorola L6 and LG VX4500 require a short pause to synchronize the script between the Go to Create Message and Send Message functions. This short pause is implemented at the Macro level in order to ensure that the script executes efficiently for as many devices as possible.

For the Motorola L6 and LG VX4500, synchronization was achieved by placing a Wait command with a short pause at the start of each Macro instance.

Notice the long key sequence for the Nokia 3200b. The key sequence for the Nokia 3200b to access the Message Alert screen is very distinct from the other two devices. In this case, a series of back commands is adopted to reach the idle screen, since the Nokia 3200b does not exit nested menus with a simple END key press.

After entering the message, typing the telephone number, and sending the message, the device executes a key sequence to clear each letter in the message and navigate through the device menus to reach the idle screen. This key sequence prepares the device to receive the Message Alert.

Reset Device

At the end of every script, it's important to return the device to the idle state. For these devices, return to idle requires a small set of key presses. Thus, a Key resource is implemented to execute the correct key sequence for each device.

9.7.4 Final Check for Send an SMS Message

Once you have finished creating a script with resources, check your script again with each device. Any Wait commands inserted into the script for one device will impact command execution for all other devices. As you check the script with different devices, you'll either need to change slot allocation or copy the script and associate each device with each copy of the script.

Change Slot Allocation

How to Change Slot Allocation:

1. In the script editor, right click the active device in the far right column. The Device menu appears.
2. Select Slot Allocation No. 1. The active device is associated to Slot 1.
3. Click Play on the toolbar to run the script.

The script runs to successful completion.

Copy a Script and Add a Device

Make sure you are in the Automation or Monitoring view with the script editor open. In this example, you've created a script with more than one device. To check the script against a particular device, you can copy and paste the script into a new file, and then add the device you want to test.

How to Copy a Script and Add a Device:

1. With the script editor open to a finished script, draw a box around all the script commands in the editor with your cursor. The commands are highlighted in black.
2. In the Edit menu, select Copy Element. The script commands are copied to the Clipboard.
3. Right click Scripts or Monitor Scripts in the Explorer.
4. Select New Script.
5. Enter the name of the new script.
6. In the Edit menu, select Paste Element. The script is pasted into a new file.
7. Right click on the device in the Device list that you want to use in the script.
8. Select Add to Script. By default the device should be added to Slot 1.

Key Points:

- An efficient script has a number of commands including checkpoints to ensure the correct command execution for each device.
- Use a WaitImage that always qualifies command execution and results. For example, for a Send an SMS Message script, make sure to check that a new message was received by verifying the Message Alert screen that alerts the user of a new text message.
- Enter plenty of notes to document the script. The more comments you add to a script, the easier it is for another user to understand your script and continue development.

10 MyDeviceAnywhere

MyDeviceAnywhere is the web interface you see once you have logged in successfully to your *DeviceAnywhere* account. *MyDeviceAnywhere* is a management and project area where you can perform the following functions:

- Review your account including subscriptions, pricing, active licenses for DeviceAnywhere Pro and DeviceAnywhere Monitoring, Trial subscriptions, and Promotions. From this page, you can edit your subscriptions or request to unsubscribe to any product on your account.
- Collaborate with your teams, partners and customers
- View test results and run reports
- Make reservations for devices
- Perform administrative tasks (when you have Admin privileges)
- And launch *DeviceAnywhere Studio*

10.1 Log in to Your *MyDeviceAnywhere* Account

Your System Admin will provide you with a username and password to log in to *MyDeviceAnywhere*.

To log in to *MyDeviceAnywhere*, follow these steps:

1. Launch your browser.
2. Enter the URL: http://www.deviceanywhere.com.
3. Click the Launch button.
4. Enter your username and password when a new login screen appears.
5. Click Login.

The *MyDeviceAnywhere* home page appears on the screen.

10.2 *MyDeviceAnywhere* Interface

Figure 10-1 shows the *MyDeviceAnywhere* home page. This area supports basic functions to manage your account and work on projects. Anything you would like to share with team members can be exported to your desktop or sent via email from here.

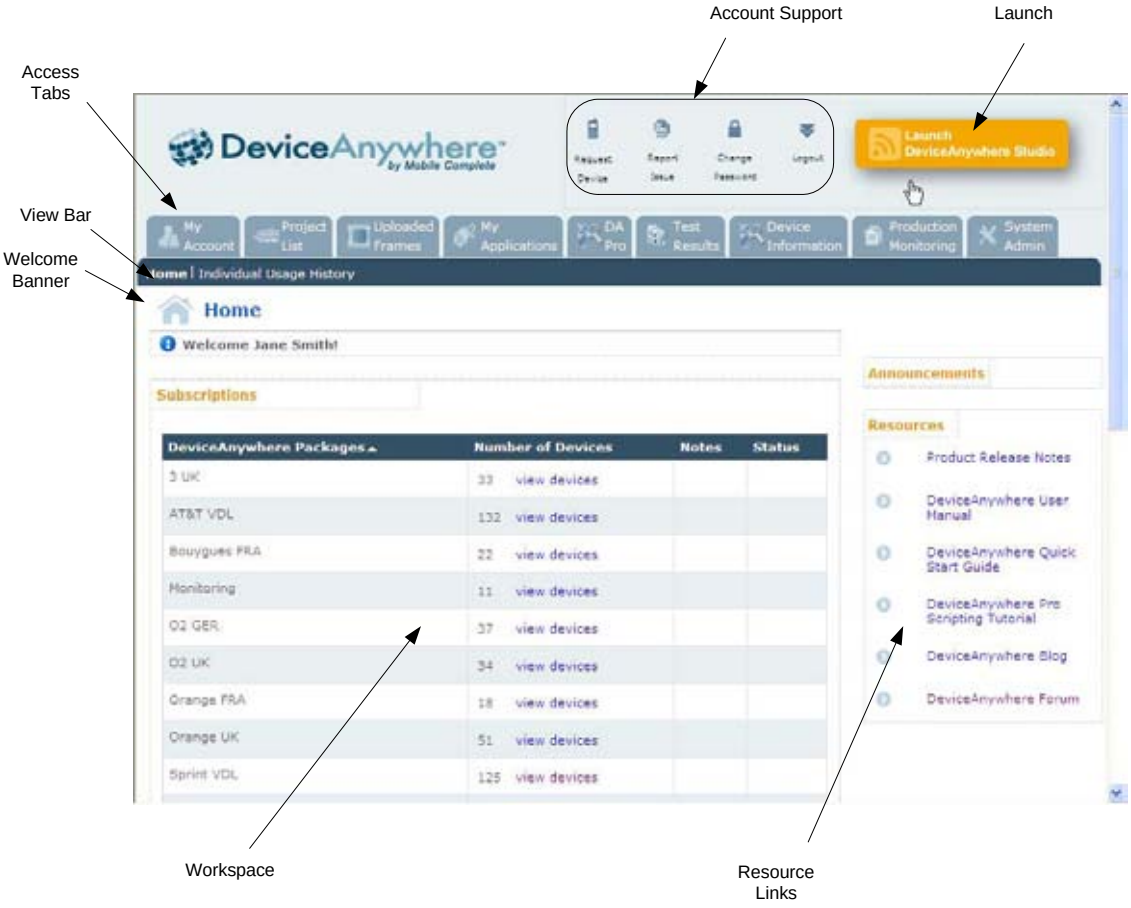


Figure 10-1 *MyDeviceAnywhere* Home Page

MyDeviceAnywhere supports three User roles—User, System Admin, and Account Admin. The privileges for each role are described below.

- User—Within a project, the User has all privileges to use licensed software features, create end-to-end scripts, test scripts, or monitoring scripts, depending on licensing. The User runs tests, uploads, and reviews test results in collaboration with team members.
- System Admin—The System Admin is responsible for setting up the workspace for Users. The System Admin creates projects, sets up User Groups, adds Users, and creates and assigns Packages (groups of devices) for projects. The System Admin is also responsible for assigning User Licences for software licences purchased by your company.
- Account Admin- The Account Admin is an executive position that is responsible for purchasing licenses and allocating resources. Account Admins are responsible for managing costs associated with one or more projects.

The Access tabs in *MyDeviceAnywhere* change based on the role assigned for each account.

The *MyDeviceAnywhere* home page has several Access tabs to manage project information. The *MyDeviceAnywhere* page layout is similar for all User, System Admin, and Account Admin accounts.

A view in *MyDeviceAnywhere* has the following components:

- Access Tabs—The Access Tabs allow you to navigate through *MyDeviceAnywhere* and accomplish many housekeeping tasks. The heading of each tab is fairly straightforward. For example, to view your projects, click on Project List. Access Tabs available to your account are based on whether you are designated as a user, Account Admin, or System Admin. Account and System Admins have additional tabs pertaining to creating projects, adding users, and supporting user accounts.
- Account Support—Account Support provides several links that let you request a new *DeviceAnywhere* handset, change your password, report a problem, or log out of *MyDeviceAnywhere*.
- View Bar—The View Bar shows the current view to help you orient yourself within the website.
- Welcome Banner— The Welcome Banner confirms you as the registered user of this *DeviceAnywhere* account.
- Resource Links—These links allow you to access the *DeviceAnywhere* User Guide and Release Notes, and participate in the *DeviceAnywhere* user community.
- Workspace—Workspace is the main work area. Here enter settings or view reports, based on activities for the current view. For example, when you click on the Test Results tab, the test results are displayed in this area.
- Launch—The Launch icon starts downloading the *DeviceAnywhere Studio* client to your desktop. After download, an account Login screen appears. Enter the user ID and password that you received from your Account Administrator.

10.3 Access Tabs for Users

This section provides a description of the Access Tabs in *MyDeviceAnywhere* for all roles—User, System Admin, and Account Admin. Refer to the tabs in your account based on the role assigned.

10.3.1 My Account

The My Account view has the following options in the View Bar:

- Subscriptions
- Account Info
- Usage History
- Invoices

Click on the options in the View Bar to open the corresponding view.

10.3.2 Subscriptions

The Subscriptions view is divided into five windows that provide the following information:

- Subscriptions—This window lists the device packages that you are subscribed to, the devices in each package, and the monthly rate for the package. You can drill down in the list to view the devices in each package.
- Monthly Hours—This window indicates the number of hours used for the month up to the present date.

- DeviceAnywhere Monitoring—This window indicates whether you are signed up for DeviceAnywhere Monitoring and shows the number of Production Monitors assigned to your account. This window also displays the Edit Subscriptions icon.
- Trials—This window shows when you are signed up for any trial licenses.
- Promotions—This window indicates when you are signed up for a free promotion. From time to time, a monthly subscription to one or more packages is available for free as a promotion. Only the hourly access fees associated with using the packages is applied to your account.

Figure 10-2 shows a list of all the subscriptions for your account.

Figure 10-2 Subscriptions in My Account

10.3.3 Editing Your Subscriptions

To edit your subscriptions, click the Edit Subscriptions icon on the My Account page. The Choose Subscriptions screen is displayed. Figure 10-3 shows the Choose Subscriptions screen where you edit the active features of your account.

At the top, the steps you take to change your account are shown on a progress bar. Each step is highlighted as you go from the selection process, confirmation, and submittal of your changes. Click Next on the bottom of the page when you want to move to the next step.

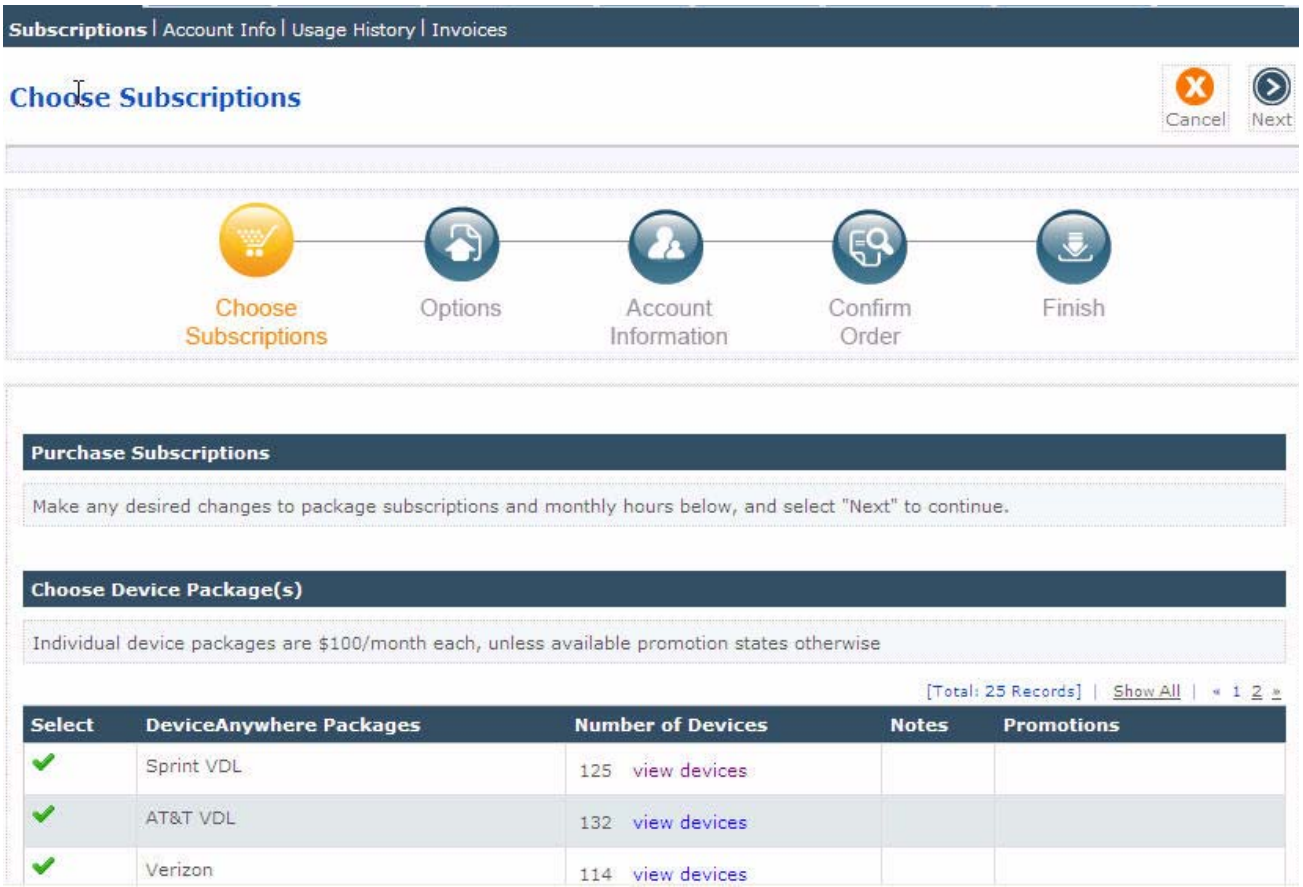


Figure 10-3 Choose Subscriptions Screen

Steps in Changing Your Account include the following:

Choose Subscriptions—This page shows your existing subscriptions. The packages that you are already subscribed to are indicated with a green check mark. When you want to add a package, click the checkbox in the Select column. To unsubscribe to a package, click the unsubscribe link. This action allows you to notify *DeviceAnywhere* to discuss your desire to unsubscribe to a feature.

Options—This page allows you to add a *DeviceAnywhere Pro* licenses to your account. Licenses are allocated in three ways: 1) The Base rate allows for two licenses; 2) Additional user licenses allows you to add in increments of 5 licenses at a time; and 3) The Platinum rate includes 24-hour priority customer support and device reservations.

Account Information—This page allows you to enter user information including your title and company name. Fields indicated with an asterisk must be completed before moving on to the next screen.

Confirm Order—This page provides a summary of the changes you have added in preceeding steps. It shows additional costs for the changes you've selected. When you want to edit the information on the page, click the back button. Go back to previous pages and change your entries. When you click Save, your order is submitted and applied to the current month.

10.3.4 Account Info

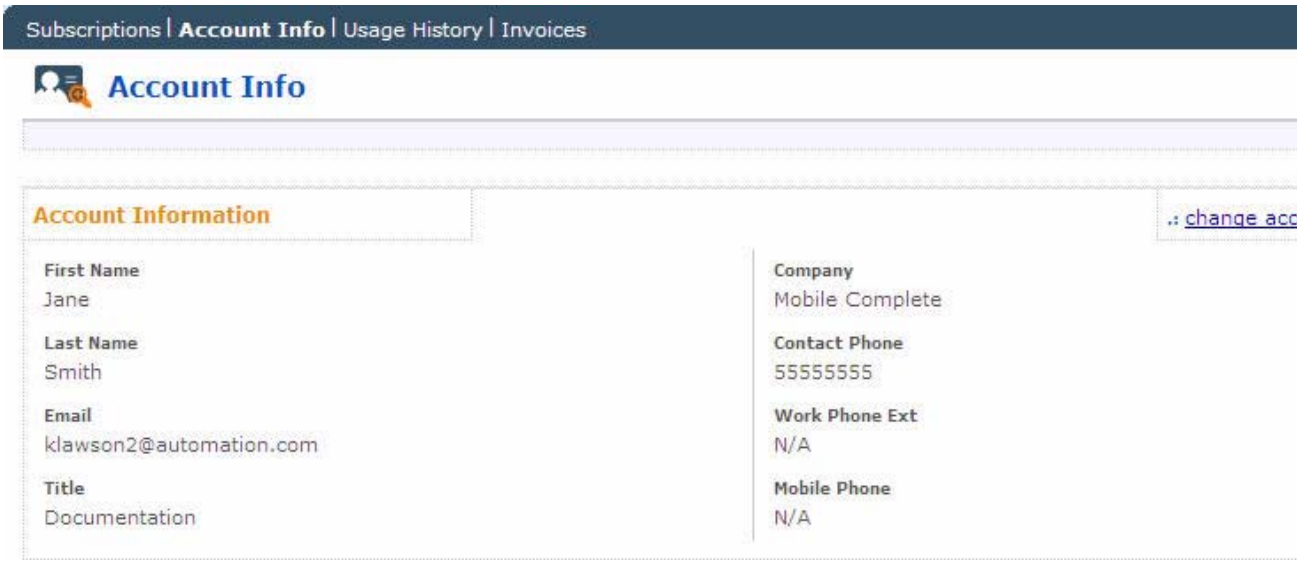
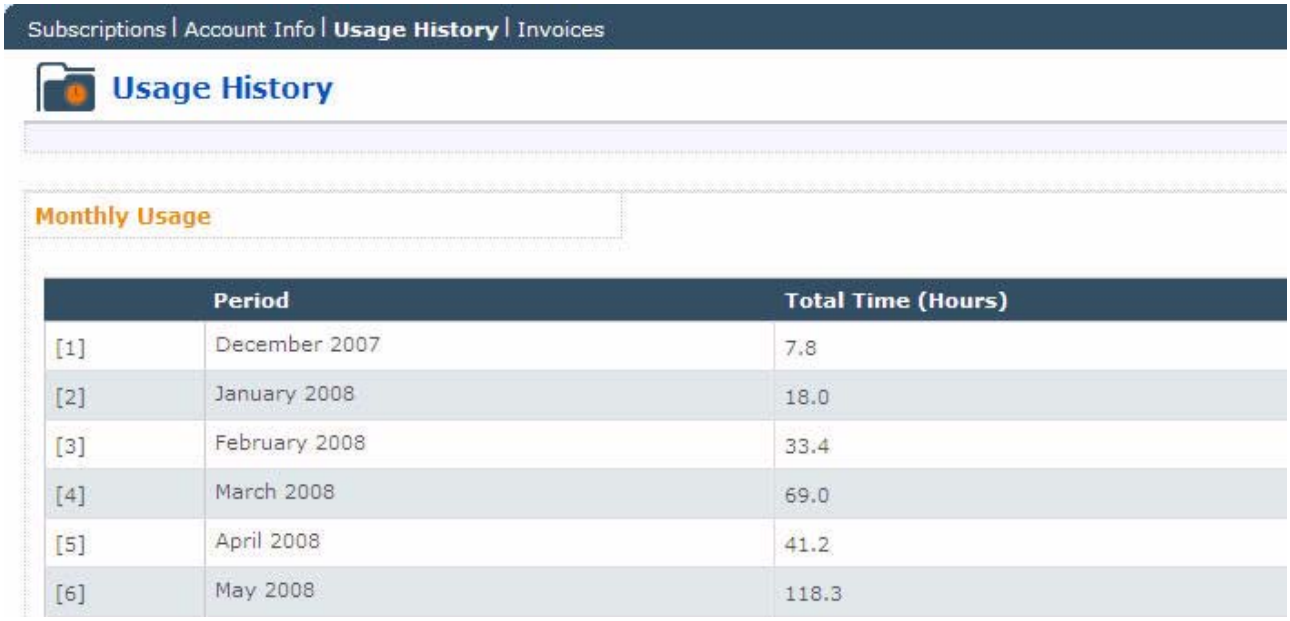


Figure 10-4 Account Information

This screen allows you to update your account information.

10.3.5 Usage History

The Usage History shows the time usage by the month. Click on the Month to open a detailed report of usage history for each User.



	Period	Total Time (Hours)
[1]	December 2007	7.8
[2]	January 2008	18.0
[3]	February 2008	33.4
[4]	March 2008	69.0
[5]	April 2008	41.2
[6]	May 2008	118.3

Figure 10-5 Usage History

10.3.6 Invoices

Click the Invoices option to view the invoices for each month.

10.3.7 Project list

The Project List view lists all the projects to which you have access. This list includes projects owned by your group; that is, projects for which you are the primary customer. The list also includes projects owned by another group, but of which you are a member.

The Project List table acts as a top-level view for drill-down. You may sort the list based on the project name, ID or creation date.

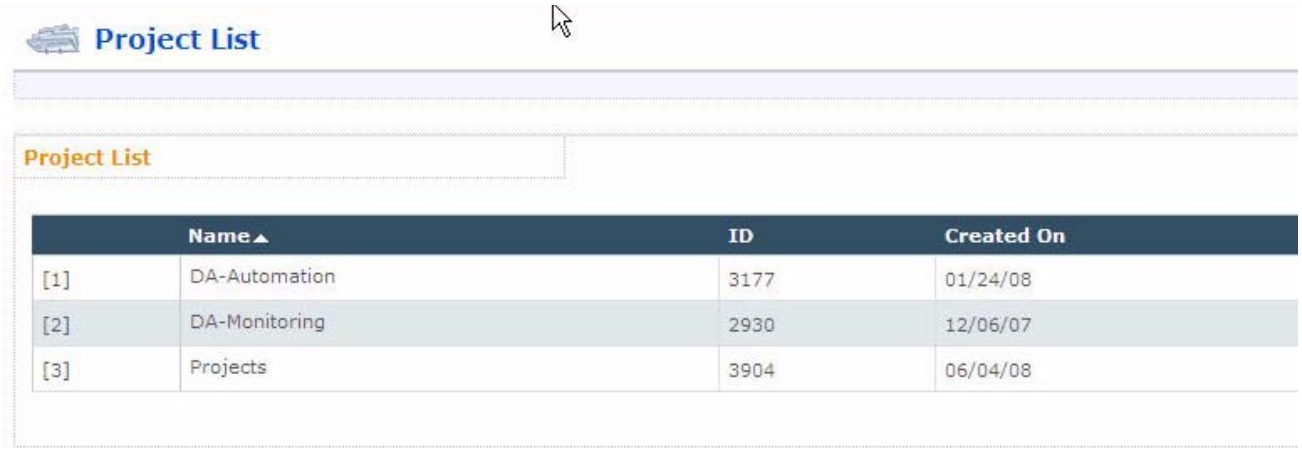


Figure 10-6 Project List

To learn more about a particular project, click on the project name in the Project List screen. Clicking on a project's name takes you to a detailed view of that project. This screen serves as a project dashboard, providing you detailed information about a particular project.

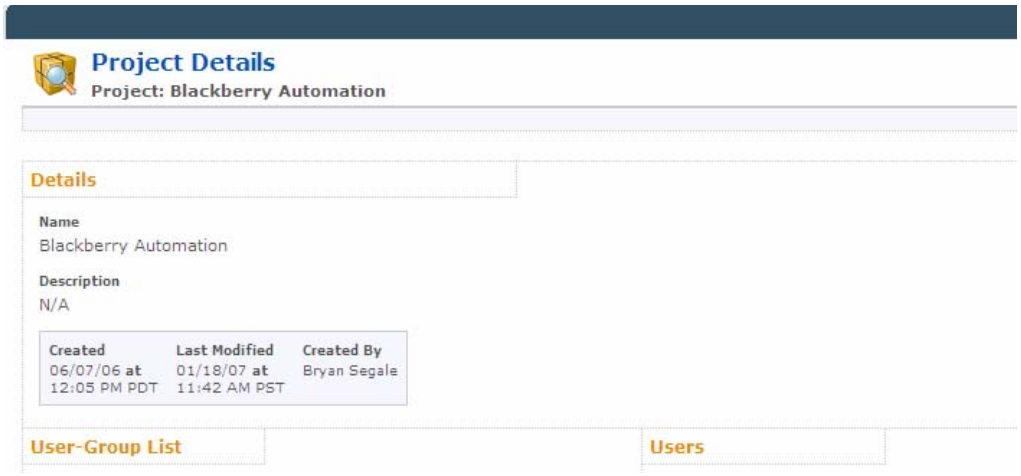


Figure 10-7 Project Details

On the Project Details page, the Details section displays basic information about a project including its name, description, date of creation and most recent modification information.

The User-Group List and Users sections displays details of all users and user groups that have access to this project. This information can include users who are not in your group, but are part of groups with whom this project is being shared.

10.3.8 Uploaded Frames

Use the Uploaded Frames tab to share and collaborate with other team members. The main page shows the list of framesets that you have uploaded.

Figure 10-8 shows the detailed results of a series of steps recorded during a live session. The uploaded frames were saved to Vodafone-Demo.

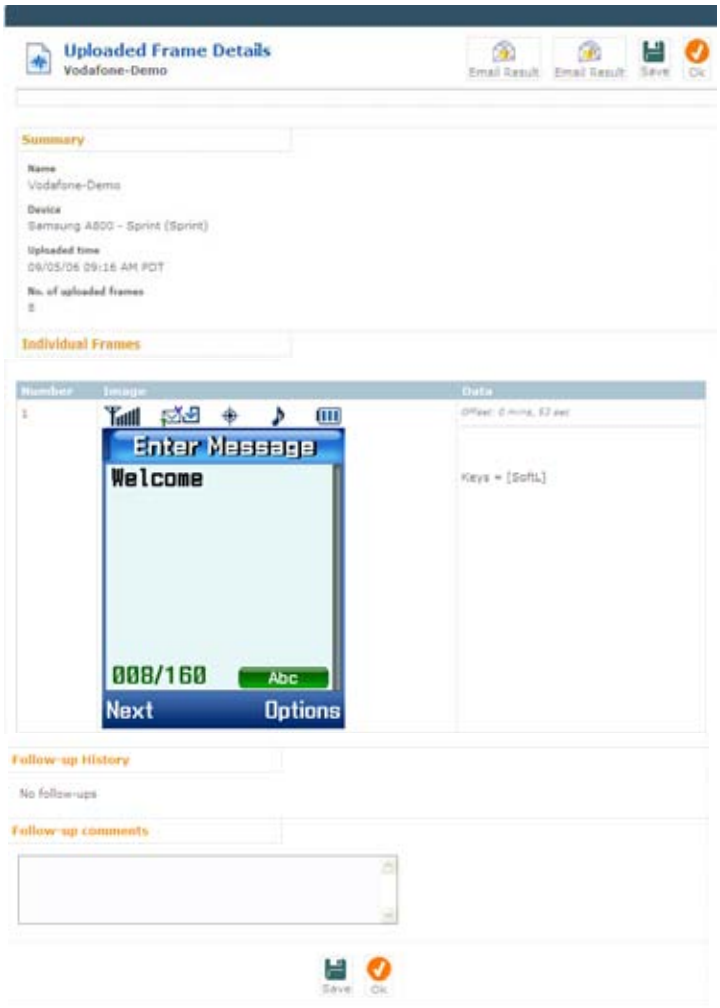


Figure 10-8 Upload Frame Details

To view frame set details (i.e. individual frames and keystrokes), click on the frame name. On this page you are able to do the following:

- View Individual Frames—Details the actual frames and their associated key presses.
- Make Comments—You can add comments to the frame sets. The comments are maintained in a comment history table displayed at the end of the page.
- Email Result—Two Email Result icons appear at the top of the screen. The first icon with a lock indicates that it is a secure email sent to registered users of *DeviceAnywhere*. These emails include a link to the Upload Frames tab in your account. The second icon indicates a public email is sent. These emails are sent to users who are not on your *DeviceAnywhere* account or to non-users. These emails contain a link that only shows the specific frame sets you have sent.

10.3.9 My Applications

illustrates the My Applications view. On this tab, a list of all applications stored on MyDeviceAnywhere is shown. The following commands are available to manage applications in your account:

- Add/Remove Files—This command allows you to Add or Remove applications from your MyDeviceAnywhere account.
- View—This command allows you to view the file name and file description of the application.
- Edit—This command allows you to edit the file name and file type for the application.
- Delete—This command allows you to delete the file.

Figure 10-9 illustrates the list of applications in the My Applications tab.l

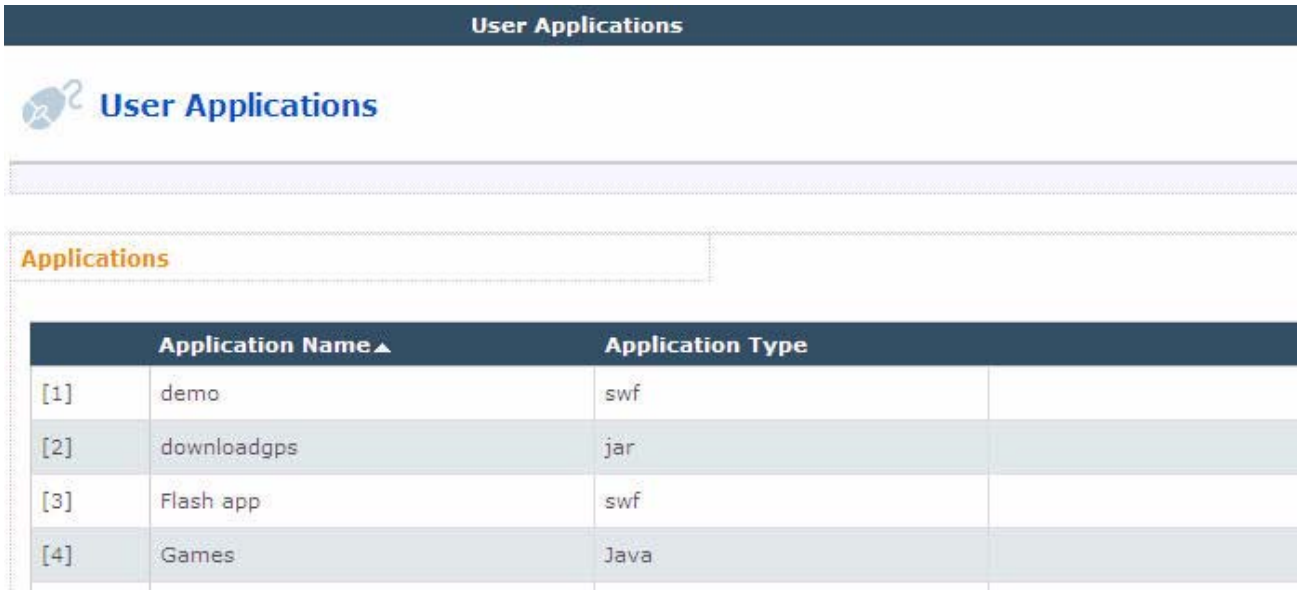


Figure 10-9 My Applications View

Upload mobile applications to your My Applications tab by following this procedure:

1. In a browser, log in to your MyDeviceAnywhere account.
2. Click the My Applications tab.
3. Click the Add Application icon at the top right of the page.
4. Enter the application name and type.
5. Click Save.
6. Click Add/Remove files.
7. Click Add File. A browser opens that allows you to navigate to the folder where the application is located on your computer.
8. Select the application for upload.
9. Click Save to upload your file.

For Java applications, you should upload jar and jad files, as well as other files required for your application.

10.3.10 Test Results

Tests that you run interactively, that are scheduled, or generated through Record and Playback or Automation scripts produce test results that are accessible through *MyDeviceAnywhere*. Each test result is a step-by-step view of what happened during the test run. Web-based access to test results allows you to view results and email results to other users.

Test results are structured as follows:

- Each run generates a test report, consisting of one or more individual results.
- Each result corresponds to a combination of one script and one device.
- Each result consists of a series of result steps that present a visual view of what happened during that test run.

The Test Results tab displays a list of all the test results in the system, for all projects to which you have access.

You can sort the list of test results based on various test attributes, and you can navigate pages by clicking on the page numbers in the top right corner. When you click on a particular result name, you can drill down to view details of that step and corresponding results.

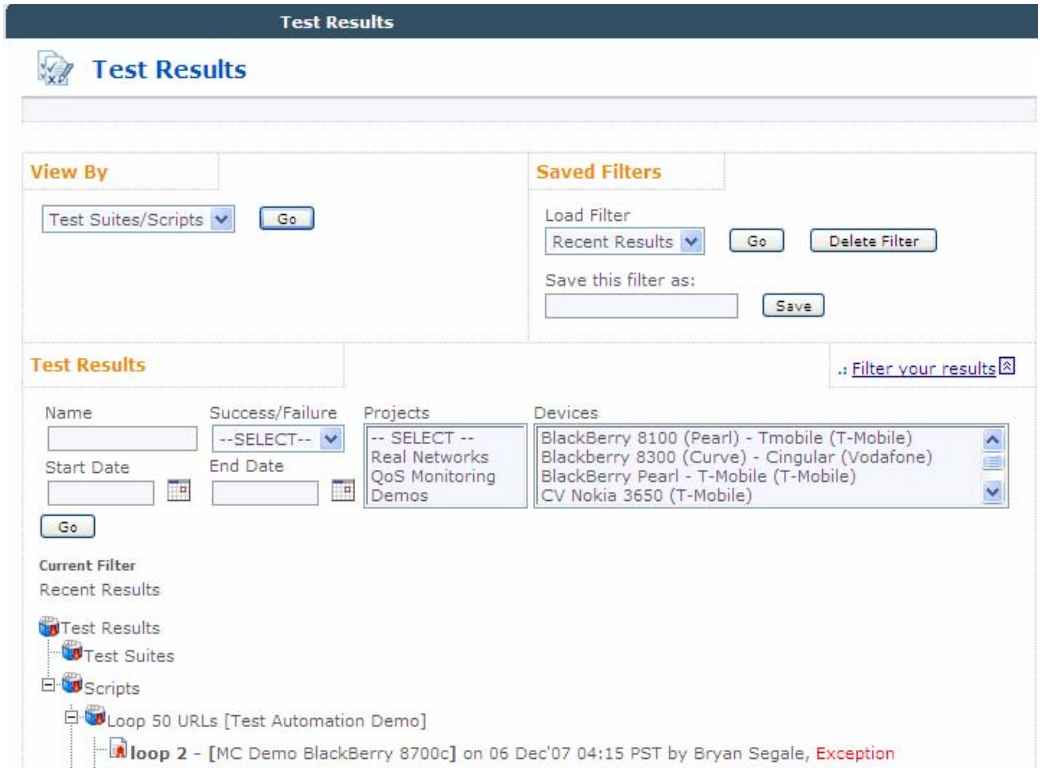


Figure 10-10 Test Results

10.3.11 Reservations

Devices can be reserved through the Reservation tab in *MyDeviceAnywhere*. Reservations are a premium service in *DeviceAnywhere Studio*. You must sign up for this service to activate the Reservations tab in *MyDeviceAnywhere*.

Reservations allow you to reserve mobile devices for usage at requested days and times. Reservations can be made up to 2 weeks in advance and for up to a one hour slot.

If another user is using the reserved device at the time that the reservation starts, a warning is issued to the user about the upcoming reservation 5 minutes before the reserved time. At the reserved time, the device is automatically released, and the reserver is notified that the device is available.

The Reservations tab supports the following options:

- My Devices allows you to see the devices associated to the project.
- Current Reservations provides an overview of the reservations you have made.
- Create Reservation allows you to create a reservation for a device.

Figure 10-11 illustrates the Create Reservation view.

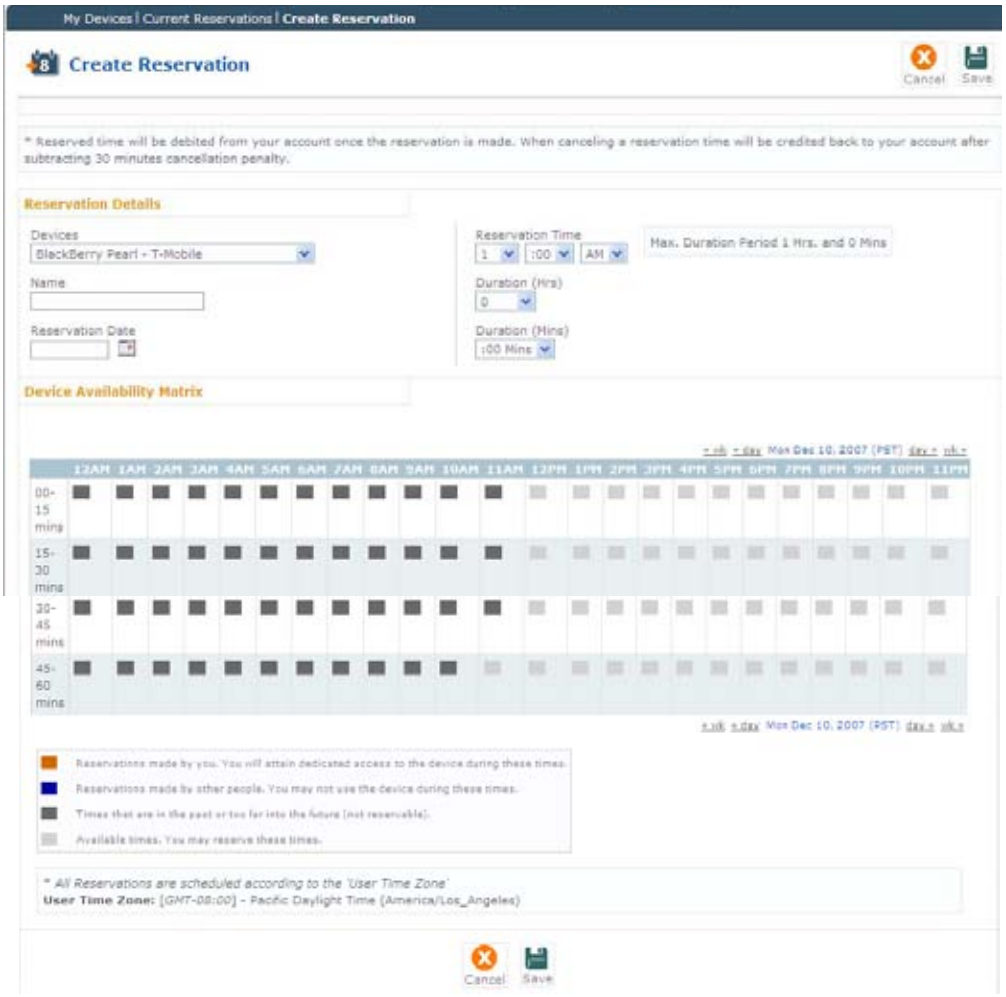

My Devices | Current Reservations | Create Reservation
Create Reservation
Cancel
Save
* Reserved time will be debited from your account once the reservation is made. When canceling a reservation time will be credited back to your account after subtracting 30 minutes cancellation penalty.
Reservation Details
Devices
BlackBerry Pearl - T-Mobile
Name
Reservation Date
Reservation Time
Max. Duration Period 1 Hrs. and 0 Mins
Duration (Hrs)
Duration (Mins)
Device Availability Matrix
12AM 1AM 2AM 3AM 4AM 5AM 6AM 7AM 8AM 9AM 10AM 11AM 12PM 1PM 2PM 3PM 4PM 5PM 6PM 7PM 8PM 9PM 10PM 11PM
00-15 mins
15-30 mins
30-45 mins
45-60 mins
Reservations made by you. You will attain dedicated access to the device during these times.
Reservations made by other people. You may not use the device during these times.
Times that are in the past or too far into the future (not reservable).
Available times. You may reserve these times.
* All Reservations are scheduled according to the 'User Time Zone'
User Time Zone: [GMT-08:00] - Pacific Daylight Time (America/Los_Angeles)
Cancel
Save

Figure 10-11 Create a Reservation

How to Create a Reservation for a Device:

1. In the *MyDeviceAnywhere* Reservation tab, select Create Reservation from the tool-bar.
2. Choose the device and carrier you want to reserve.
3. Choose the reservation time. The Device Availability Matrix shows you what times are available for the device.
4. Click Save to create a reservation.

How to Cancel a Reservation for a Device:

1. In the *MyDeviceAnywhere* Reservation tab, select Current Reservations from the tool-bar. All reservations are displayed.
2. Click cancel on the same line for the reservation you want to cancel.
3. Click the Save icon.

When the reservation is saved, you can view your reservations by clicking the Current Reservations link in the tool bar. Reserved time is debited from your account once the reservation is made.

When canceling a reservation, time is credited back to your account after subtracting the maximum 30 minutes cancellation penalty.

10.3.12 Device Search

Device Search allows you to search for devices by package, manufacturer, and model, similar to the Device Search function on the DeviceAnywhere.com website. Additionally, the Device Search in *MyDeviceAnywhere* allows you to export search results to an Excel file.

You can view devices both with icons and without icons (which results in faster search results).

Figure 10-12 shows a Device Search without icons.

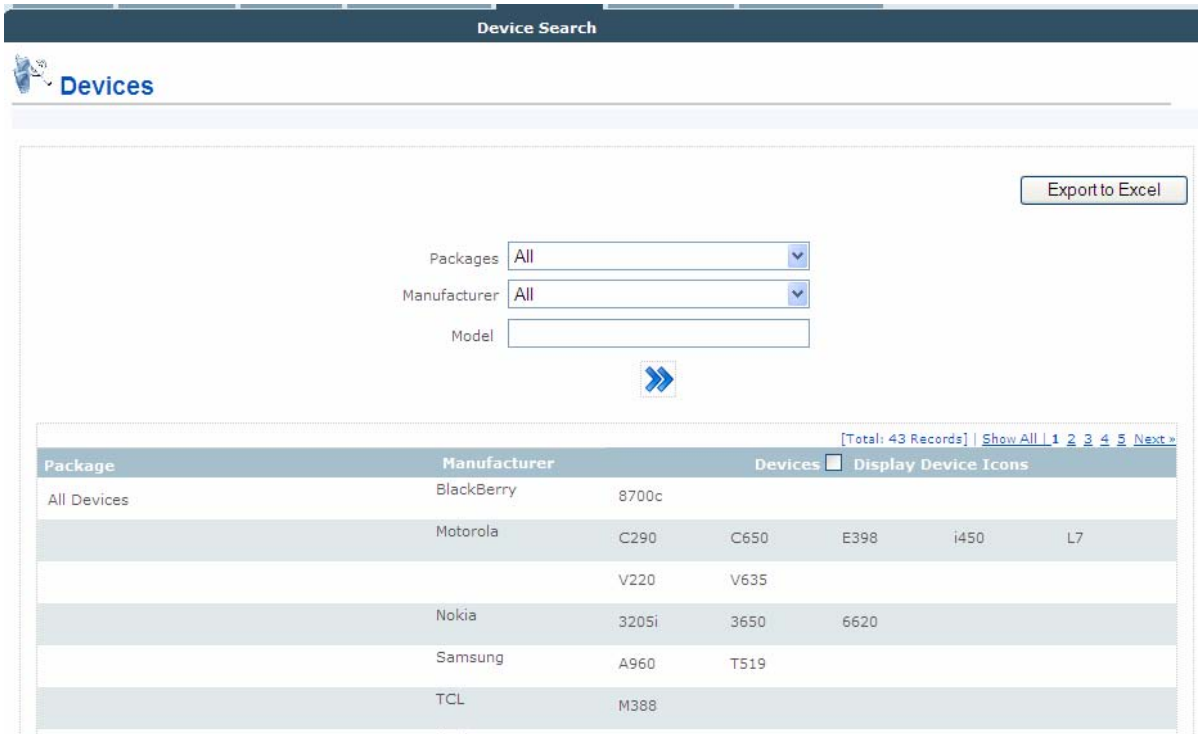


Figure 10-12 Device Search Results without Icons

You can also search across all packages for a specific device model; the results lists the packages which contain that model. In the below example, the Motorola K1 search results are shown.

Figure 10-13 shows the results of a search across all packages for the Motorola K1 device.

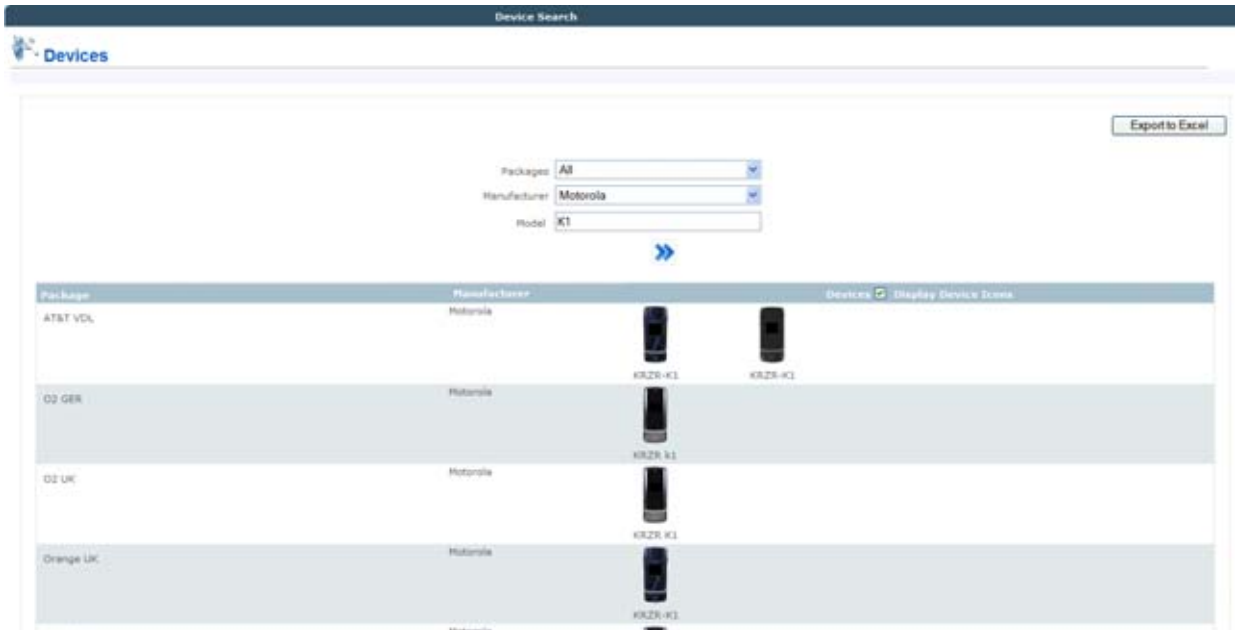


Figure 10-13 Motorola K1 Search Results

10.3.13 Production Monitoring

The Production Monitoring view provides a list of all the Production Monitors to which you have access. This tab is available when you have a license for *DeviceAnywhere Monitoring* and you are assigned to a license by your System Admin.

Figure 10-14 illustrates Production Monitoring with View Data displayed.

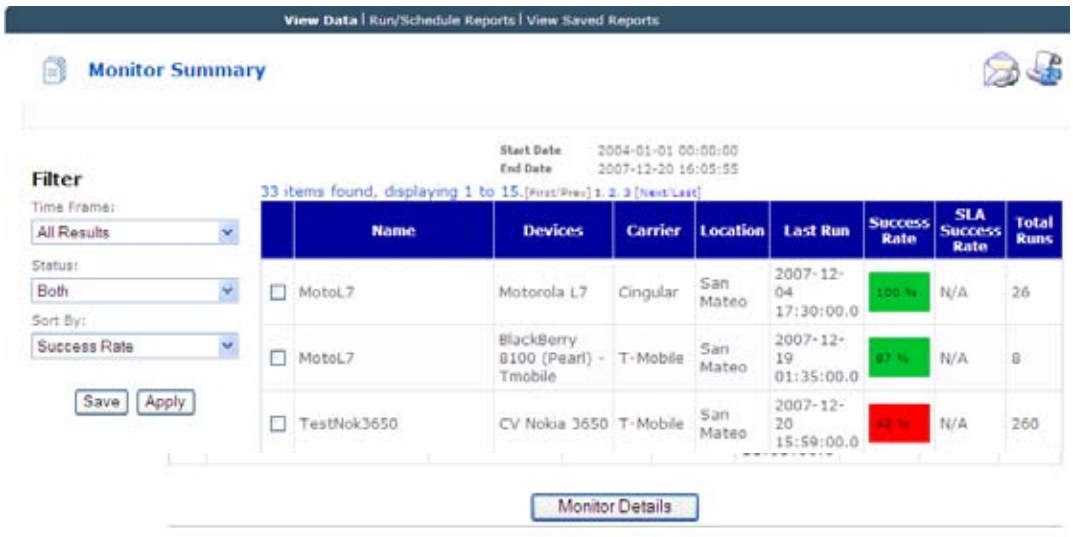


Figure 10-14 Production Monitoring - View Data

The Production Monitoring view has the following options:

- View Data
- Run/Schedule Reports
- View Saved Reports

View Data

When you click on Production Monitoring, you are sent to the View Data tab by default. This page is the Monitor Summary view. In this view, you can do the following:

- Run ad hoc reports
- Choose and save filter criteria
- Save the reports you generated in this view
- Export the results to excel
- Email the results

In this section, you can run ad hoc reports. You are also able to choose filters and choose views (types and levels of reports) and generate reports. You can save the criteria (for future reference) you used to generate a report by clicking Save Criteria. After running a report, you can save the results by clicking Save Reports.

Run/Schedule Reports

When you save filter criteria by clicking Save Criteria in the View Data page, the filter settings are saved here. From this area, you can choose from a list of all the saved criteria that you have defined previously, and generate new reports based on those criteria. You can also schedule certain saved criteria to run on a regular schedule and post the saved reports (or email them). You have the option to create new filters by editing the saved criteria from previous runs.

View Saved Reports

The View Saved Reports link lets you manage all saved reports, those that were scheduled in both View Data and Run/Schedule Reports.This page allows you to View or email the saved reports.

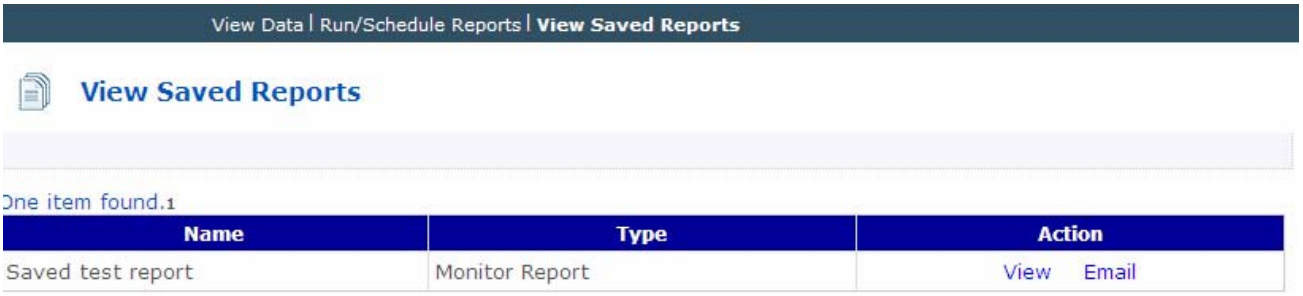


Figure 10-15 Production Monitoring - View Saved Reports

10.4 Access Tabs for System Admin

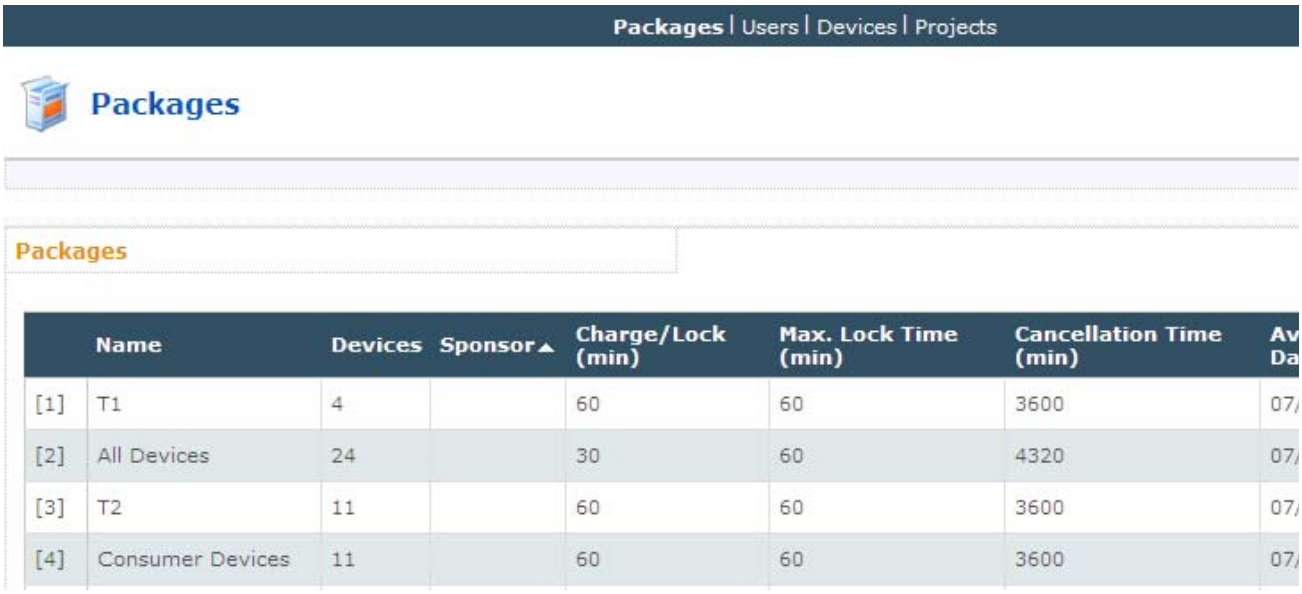
The DeviceAnywhere is an application that can be used across many groups within an organization. To allow many people on different project teams to work with the system, but keep project work separate from each other, DeviceAnywhere has an integrated set of administrative functions that allow an Administrator to manage users, user groups, device allocations (packages) and projects. The System Admin tab supports all the user account functions.

In this section, you get a brief introduction to the administrative tools available in the DeviceAnywhere.

The toolbar under the System Admin tab provides the following options:

- Packages
- Users
- Devices
- Projects

Figure 10-16 illustrates the System Admin tab. By clicking the Packages link on the toolbar, all the Packages created by the System Admin are listed.



	Name	Devices	Sponsor	Charge/Lock (min)	Max. Lock Time (min)	Cancellation Time (min)	Av Da
[1]	T1	4		60	60	3600	07/
[2]	All Devices	24		30	60	4320	07/
[3]	T2	11		60	60	3600	07/
[4]	Consumer Devices	11		60	60	3600	07/

Figure 10-16 System Admin Tab with Packages Page Open

10.4.1 Add a Package

In a browser, navigate to the home page of the *MyDeviceAnywhere* web site.

1. Log in as an Admin user.
2. Click on the System Admin tab.
3. Click on the Packages link in the toolbar. All existing Packages are displayed.
4. Click Add Package.
5. Enter the name of the new package. This name reflects the project and devices that will be used.
6. Enter the Device Reservation Policy.
7. Click Save.

The new Package is created. However, you need to assign devices to it based on project requirements.

Device Reservation Policy

Your Device Reservation Policy has 5 settings:

Availability Date—

Min Charge per lock—

Cancellation period—

Min time between locks—

These settings are governed by the Account Admin. Contact the Account Admin for more information.

10.4.2 Add Devices to a Package

In a browser, navigate to the home page of the *MyDeviceAnywhere* web site.

1. Log in as an Admin user.
2. Click on the System Admin tab.
3. Click on the Packages link in the toolbar. All existing Packages are displayed.
4. Click Device Distribution on the top right corner of the screen. The Device Distribution screen is displayed.
5. Select the Package name from the drop-down menu.
6. Check the Set on live dates on devices checkbox.
7. User Override previous on live dates when you are editing Package details.
8. Under the Available Devices column, double click on devices to move them over to the Already Provisioned column. These are the devices assigned to the Package.
9. Click Save.

The Package is now defined in *MyDeviceAnywhere*. Figure 10-17 shows devices moved from Available Devices to the Already provisioned column to create a new Package.

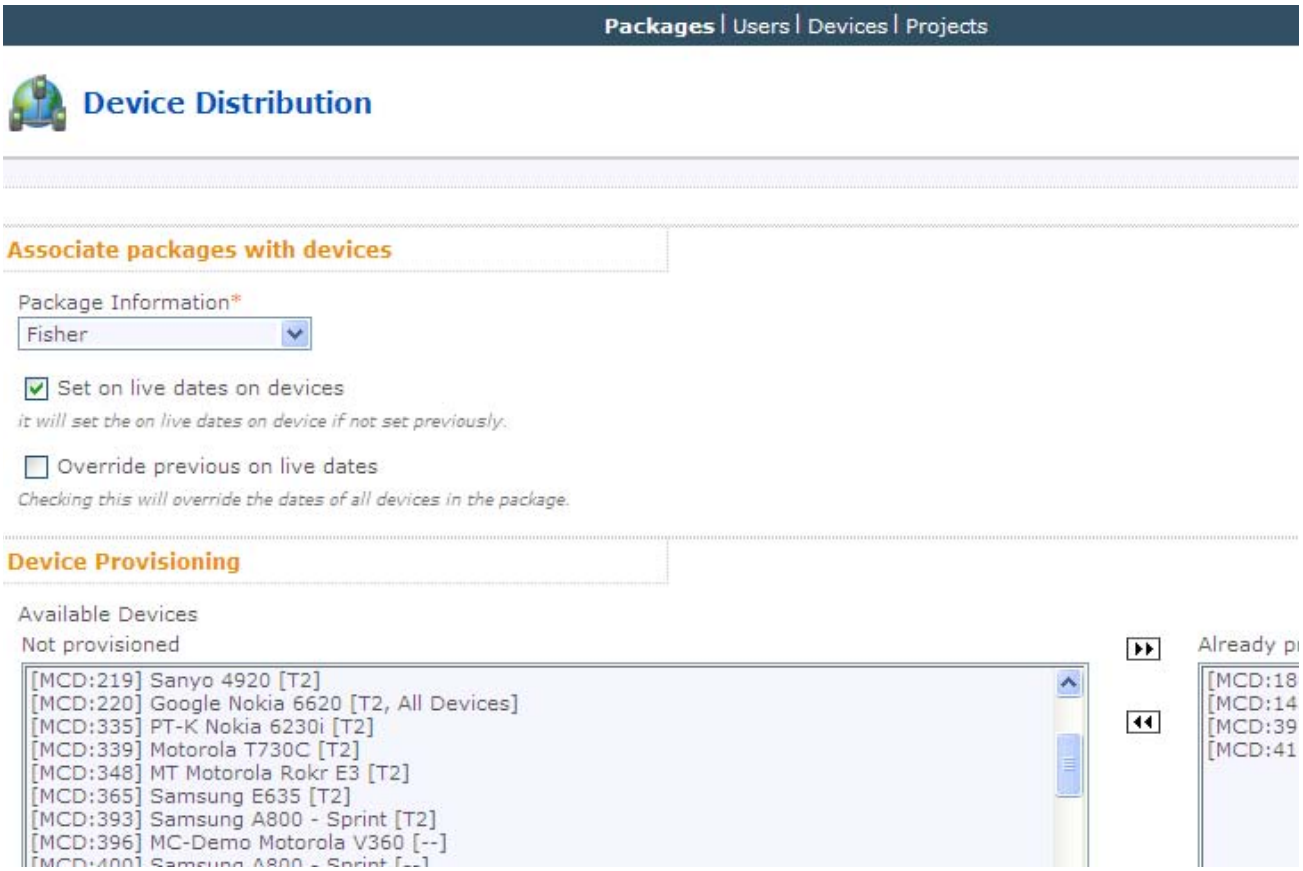

Packages | Users | Devices | Projects
Device Distribution
Associate packages with devices
Package Information*
Fisher
Set on live dates on devices
it will set the on live dates on device if not set previously.
Override previous on live dates
Checking this will override the dates of all devices in the package.
Device Provisioning
Available Devices
Not provisioned
[MCD:219] Sanyo 4920 [T2]
[MCD:220] Google Nokia 6620 [T2, All Devices]
[MCD:335] PT-K Nokia 6230i [T2]
[MCD:339] Motorola T730C [T2]
[MCD:348] MT Motorola Rokr E3 [T2]
[MCD:365] Samsung E635 [T2]
[MCD:393] Samsung A800 - Sprint [T2]
[MCD:396] MC-Demo Motorola V360 [--]
Already p
[MCD:18
[MCD:14
[MCD:39
[MCD:41

Figure 10-17 Device Distribution Screen

Devices are grouped into packages and assigned to a user group. Because you may have many users and projects going on in parallel, it is important to be able to group devices into packages and then assign those packages to user groups. Having multiple packages allows you to assign devices to a relevant user group. Not all groups need access to all devices.

10.4.3 Create a User Group

A User Group is a group of individuals who are assigned to a particular project. It is practical to assign privileges or access to resources to an entire User Group rather than assigning individuals. This feature allows you to manage at the group level.

Figure 10-18 illustrates the Create a User Group screen.

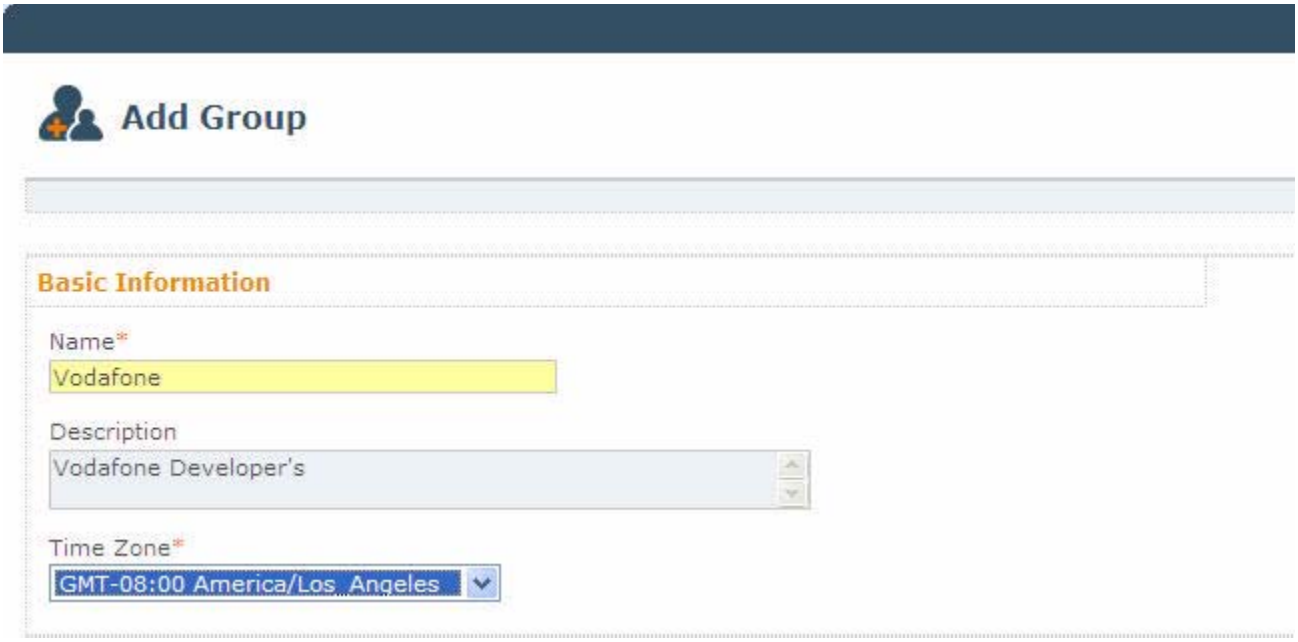


Figure 10-18 Create a User Group

In a browser, navigate to the home page of the *MyDeviceAnywhere* web site.

1. Log in as an Admin user.
2. Click the System Admin tab.
3. Click the Add User Group icon near the top right of the page.
4. Enter a name for the new User Group.
5. Enter a description for the group.
6. Select the Time zone of your location from the drop-down menu.
7. Add yourself to the user group by moving your username to the Assigned column User Group Added.
8. Double click on user names to move them to the Assigned column.
9. Double click on the Project name for this group.
10. Click the Save when you are done assigning Users and Projects.

A user group can be used to assign privileges at the group level.

10.4.4 Add a User

Figure 10-19 shows the Add a User screen where User information is entered into the system.

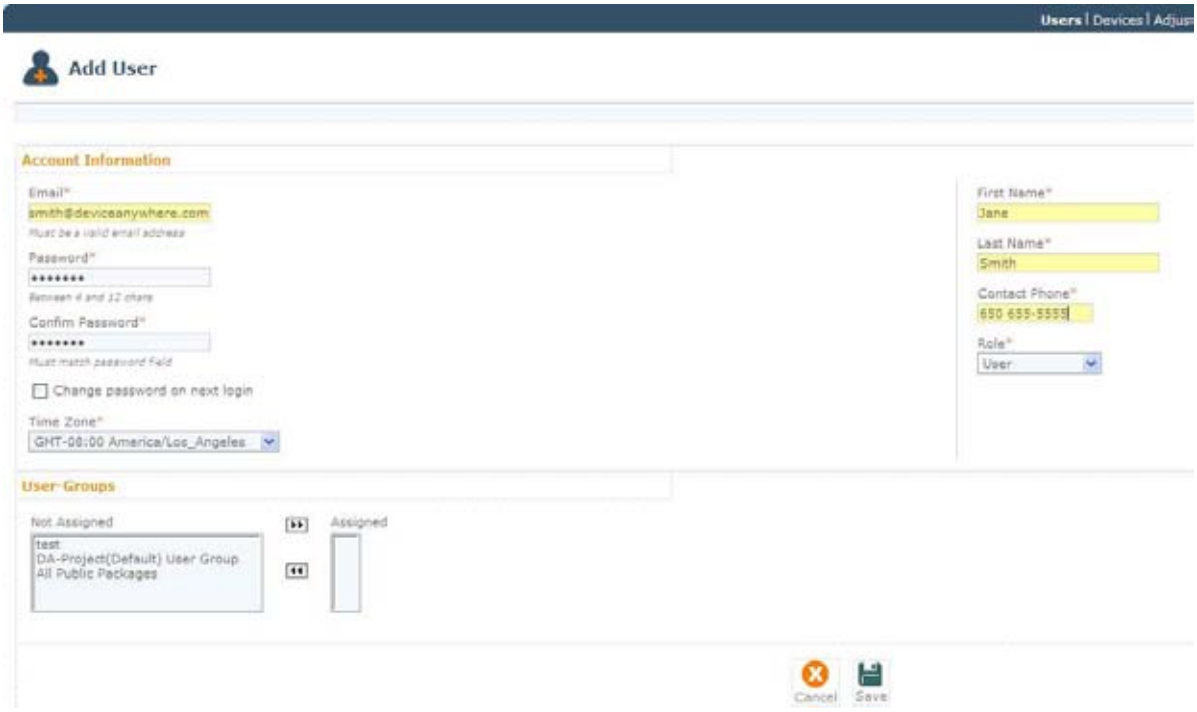


Figure 10-19 Add a New User

In a browser, navigate to the home page of the *MyDeviceAnywhere* web site.

1. Log in as an Admin user.
2. Click the System Admin tab.
3. Click the Add User icon near the top right of the page. The User page is displayed.
4. Enter the information for the User such as name, email address, phone number, and so on.
5. Select the User Role: User, System Admin, or Account Admin. Most accounts are created for basic Users.
6. Click the Save icon to add the new User.

System Admin contains views for the administration of users, devices and projects. Only Admins and System Admins have access to this tab.

When adding a user, you must choose a Role. The options are User, Admin, or Account Admin. Account Admins are the highest level of user, with access to all *MyDeviceAnywhere* functionality. Only Account Admins may make changes to the account, including adding or changing subscriptions and editing credit card information. Admins (and Account Admins) can create new users, projects (used for managing Record and Playback scripts), and edit allowances.

10.4.5 Devices

Use the Devices Admin page to monitor and edit devices, make reservations and manage reservations. Through this tab, you can view all devices in your packages to see which are in use and who is using them, and you can view and create reservations. For more information on creating reservations.

10.4.6 Add a Project

Figure 10-20 shows the Add Project screen.

Figure 10-20 Add Project Screen

In a browser, navigate to the *My DeviceAnywhere* home page.

1. Login as an Admin user.
2. Click the System Admin tab.
3. Click the Projects link just below the System Admin tab.
4. Click the Add Project icon near the top right of the page.
5. Name the new project and enter a description for it. Assign users or a user group to the project.
6. Click Save to create the new project.

10.4.7 DA Pro

The DA Pro page allows the System or Account Admin to allocate DeviceAnywhere Pro licenses to Users in a project.

10.5 Assign *DeviceAnywhere Pro* Licenses

In a browser, navigate to the *MyDeviceAnywhere* home page.

1. Click on the DA Pro tab. The Manage DA Pro Licenses view is displayed. At the top, the number of licenses purchased is shown.
2. Under the License Assigned column, click the checkbox for the users who will use a *DeviceAnywhere Pro* license.
3. Click OK.

Figure 10-21 shows the User List indicating user who are assigned a DA Pro license.

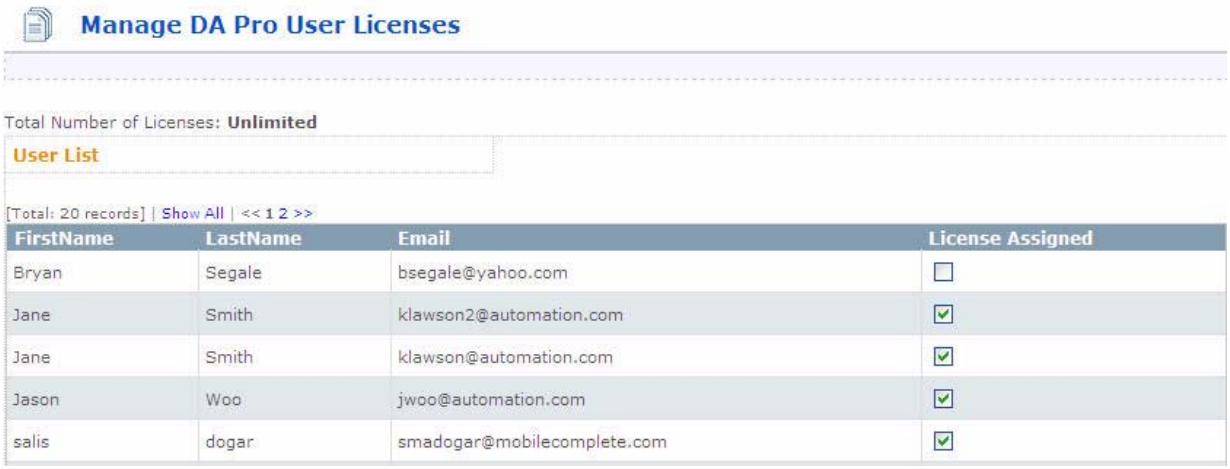


Figure 10-21 DA Pro License Allocation

10.6 Access Tabs for Account Admin

Account Admin contains views for the administration and viewing of Account Profile and Usage History. Only Account Admins can make changes to the account, including adding or changing subscriptions and editing credit card information.

10.6.1 Account Profile

The Account Profile page shows the account information of the current user. The account admin can edit the Account Profile (Company Info and Account contacts) using the Edit icon.

The Account Profile page has two windows:1) Company Info, and 2) the Primary contact information.

Figure 10-22 shows the Account Profile page in Edit mode.

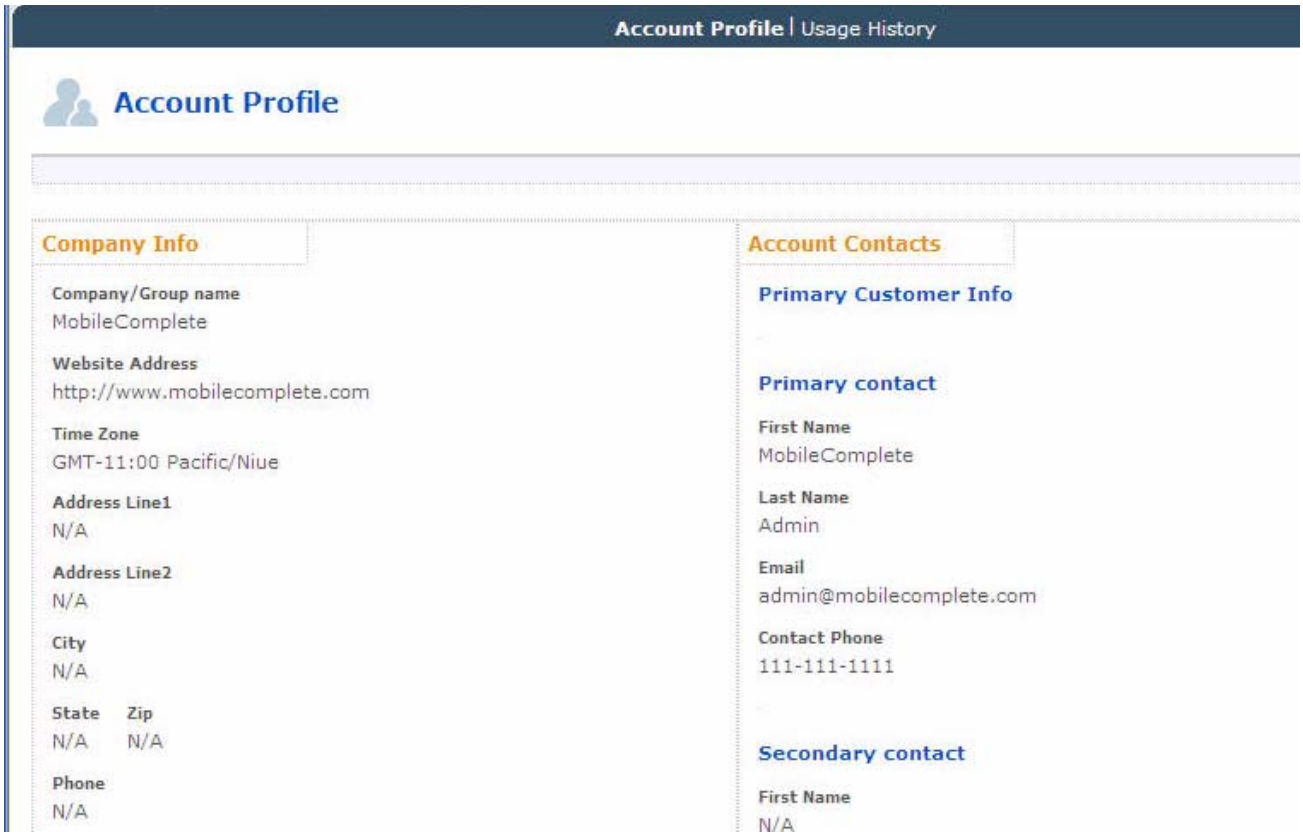


Figure 10-22 Account Profile

10.6.2 Usage History

The Usage History view shows usage history of the entire account. Clicking on an individual month shows detailed reports by user and device.

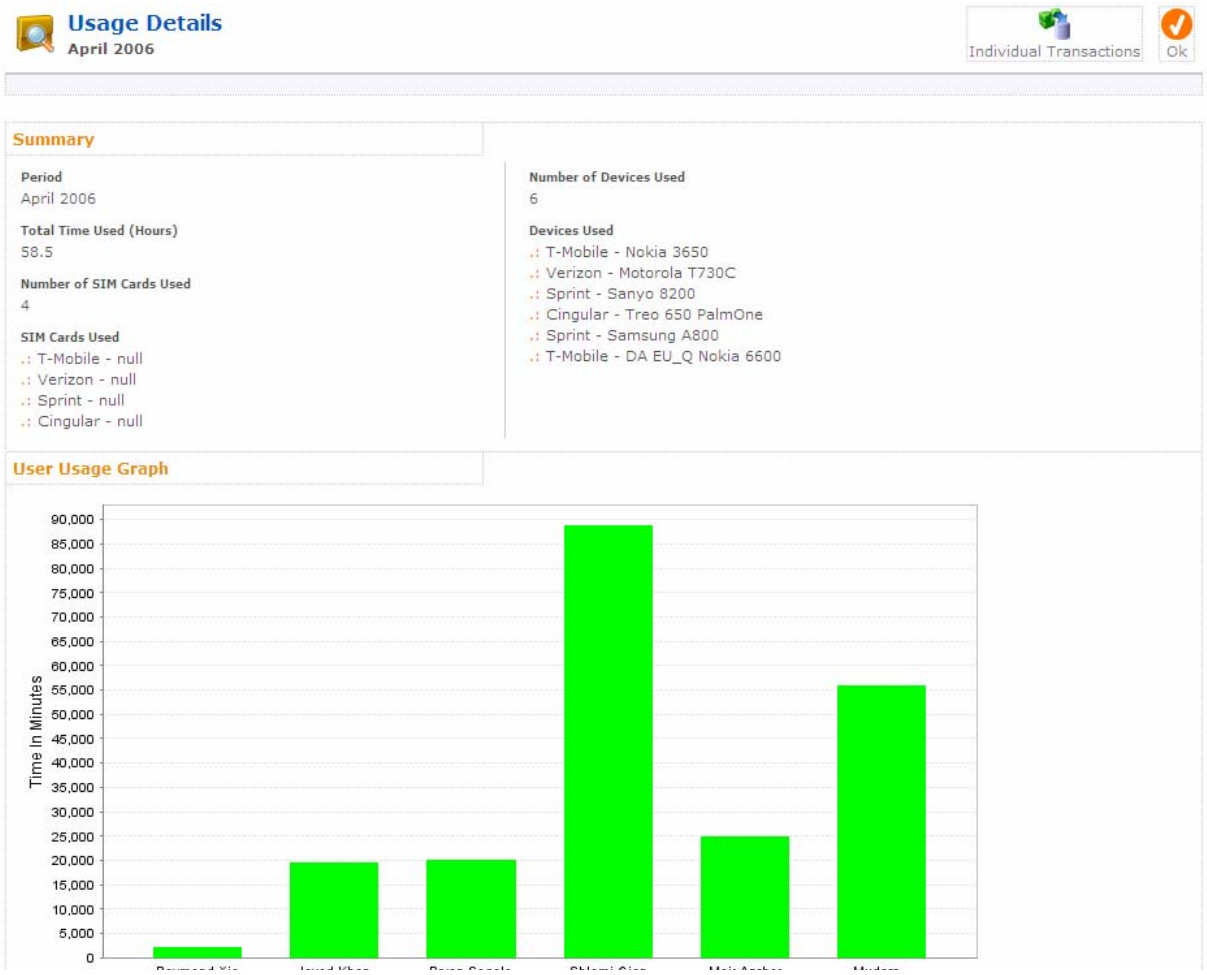


Figure 10-23 Detailed Account Usage by User